FACTORS CONTRIBUTING TO INFORMATION TECHNOLOGY SOFTWARE PROJECT RISK : PERCEPTIONS OF SOFTWARE PRACTITIONERS

FAIZUL AZLI MOHD RAHIM

JANUARY 2011

FACTORS CONTRIBUTING TO INFORMATION TECHNOLOGY SOFTWARE PROJECT RISK : PERCEPTIONS OF SOFTWARE PRACTITIONERS

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of
Doctor of Philosophy

By

FAIZUL AZLI MOHD RAHIM

JANUARY 2011

DECLARATION

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification at this or any other university or other institution of learning.

In the name of Allah,

The most gracious,

The most merciful,

All praises and thanks to Allah

ACKNOWLEDGEMENT

Thank you and gratitude to my supervisor, Dr. Abdul Halim Boussabaine for the guidance and criticisms provided throughout the study.

I adore my wife, my children and my families for their boundless love, their prayers and their support.

Grateful to my sponsors, friends and colleagues.

ABSTRACT

The majority of software development projects normally connected with the application of new or advanced technologies. The use of advanced and, in most cases, unproven technology on software development projects could leads to a large number of risks. Every aspect of a software development project could be influenced by risks that could cause project failure. Generally, the success of a software development project is directly connected with the involved risk, i.e. project risks should be successfully mitigated in order to finish a software development project within the budget allocated.

One of the early researches on risk of software projects was conducted by Boehm (1991) where the research identified top 10 risk factors for software project. Boehm research had been the starting point of research in risk of software projects. For the past 10-15 years, many researches had been conducted with the introduction of frameworks and guidelines. However, still software development project failures had been reported in the academic literatures. Researchers and practitioners in this area have long been concerned with the difficulties of managing the risks relating to the development and implementation of IT software projects.

This research is concerned specifically with the risk of failure of IT software projects, and how related risk constructs are framed. Extant research highlights the need for further research into how a theoretically coherent risk construct can be combined with an empirical validation of the links between risk likelihood, risk impact on cost overrun, and evidence of strategic responses in terms of risk mitigation

The proposal within this research is to address this aspect of the debate by seeking to clarify the role of a project life cycle as a frame of reference that contracting parties might agree upon and which should act as the basis for the project risk construct. The focus on the project life cycle as a risk frame of reference potentially leads to a common, practical view of the (multi) dimensionality setting of risk within which risk factors may be identified and which believe to be groundéd across a wide range of projects and, specifically in this research, for IT software projects.

The research surveyed and examine the opinions of professionals in IT and software companies. We assess which risk factors are most likely to occur in IT software projects; evaluate risk impact by assessing which risk factors IT professionals specifically think are likely to give rise to the possibility of cost overruns; and we empirically link which risk mitigation strategies are most likely to be employed in practice as a response to the risks and impacts identified.

The data obtained were processed, analysed and ranked. By using the EXCEL and SPSS for factor analysis, all the risk factors were reduced and groups into clusters and components for further correlation analysis. The analysis was able to evidence opinion on risk likelihood, the impact of the risk of cost overrun, and the strategic responses that are likely to be effective in mitigating the risks that emerge in IT software projects.

The analysis indicates that it is possible to identify a grouping of risk that is reflective of the different stages of the project life cycle which suggest three identifiable groups when viewing risk from the likelihood of occurrence and three identifiable groups from a cost overrun perspective. The evidence relating to the cost overrun view of risk provided a stronger view of which components of risk were important compared with risk likelihood. The research account for this difference by suggesting that a more coherent framework, or risk construct, offered by viewing risk within the context of a project life cycle allows those involved in IT software projects to have a clearer view of the relationships between risk factors. It also allows the various risk components and the associated emergent clusters to be more readily identifiable.

The research on strategic response indicated different strategies as being effective between risk likelihood versus cost overrun. The study was able to verify the effective mitigation strategies that are correlated to the risk components. In this way, the actions or consequences conditioned can be observed on identification of risk likelihood and risk impact on cost overrun. However, the focus of attention on technical issues and the degree to which they attract strategic response is a new finding in addition to the usual reports concerning the importance of non-technical aspects of IT software projects.

The research also developed a fuzzy theory based model to assist software practitioners in the software development life cycle. This model could help the practitioners in the decision making of dealing with risks in the software project.

The contribution of the research relates to the assessment of risk within a construct that is defined in the context of a fairly broadly accepted view of the life cycle of projects. The software risk construct based on the project management framework proposed in this research could facilitates a focus on roles and responsibilities, and allows for the coordination and integration of activities for regular monitoring and aligning with the project goals. This contribution would better enable management to identify and manage risk as they emerge with project stages and more closely reflect project activity and processes and facilitate the risk management strategies exercise.

Keywords: risk management, project planning, IT implementation, project life cycle

Table of Contents

List of Abbreviations

IT	Information technology
SPSS	Statistical Package for Social Sciences
ANOVA	Analysis of variance
KMO	Kaiser-Meyer-Olkin
PMI	Project Management Institute
PMBOK	Project Management Body of Knowledge
IS	Information system
PCA	Principal component analysis
SDLC	Software development life cycle
PRAM	Project risk analysis and management
APM	Associations of Project Managers
PM	Project manager
HSD	Honestly Significance Difference
CMM	Capability Maturity Model
ANN	Artificial Neural Network
FST	Fuzzy set theory
FDSS	Fuzzy decision support system
MBF	Membership function

CHAPTER 1 INTRODUCTION

1.1. Introduction

The management of IT projects has raised some important challenges for both the theory and practice of public and private sector management. Many high profile examples exist of cost overruns and service delivery failures around 'new' projects where outcomes are not 'as planned'. Examples of the huge scale and wide scope of such failures are not hard to find (see the Table 1.1 in Appendixes, of examples in Charette (2005) which provides some startling reading). The ways in which managers deal with both the inherent uncertainty and the complexity of such projects and the manner in which error cost is embedded within decision making remain major challenges for organisations and ones that have been seen to embed the potential for 'crisis' within the organisation's strategy (Smith, 1995).

Project management serves as a useful context to illustrate the ways in which organisations 'fail' to achieve the objectives that they set for themselves. An important issue within projects concerns the manner in which the unique and emergent characteristics of the project generate risks and the implications that this has for managerial decision-making. A key element of this process has related to the manner in which managers within organisations perceive the nature of the risks they face (Coles & Hodgkinson, 2008) and how these perceptions shape their 'knowledge' and understanding (Seidl, 2007). As these perceptions of risk will shape the manner in which organisational defences are configured then they form an important part of the process by which controls are put in place.

The manner in which such uncertainty has been incorporated into decision making has attracted a considerable amount of attention within the literature (Collingridge, 1984, 1992; Handy, 1994, 1995; Reason, 1997) and has resulted in some observers comparing the associated failures of managerial processes as akin to 'sloppy management' (Turner, 1994). The complexity that is inherent within projects exposes the weaknesses that exist in management processes and functions and results in an inability of managers to deal with the emergent properties that are inherent in all major projects. In this regard, managerial perceptions and'understanding' will shape their interpretations of emergent properties and may result in processes of cognitive blindness (Coles & Hodgkinson, 2008).

Projects are also 'multi-level issues' (Dansereau and Yammarino. 2002) and are resistant to simple solutions, and attempts to impose forms of 'template management' that have proved successful elsewhere (Smith, 2005). As such, they require organisations to consider the base principles on which both the decisions and the manner in which the task demands generate challenges around the cohesion and control of organisational activities. Projects also require organisations to reflect on the ways in which they frame, manage and disseminate the knowledge that is required to make sense of the complex nature of the project to those who will manage it. This is a particularly important issue when dealing with external sources of risk (Palomo, Insua & Ruggeri, 2007) and the manner in which such risks can impact on the performance of the project.

The management of a project should not be seen as an additional part of the usual day to day process of running a business, but a more serious, thorough, dynamic and practical approach should be taken strategically in dealing with the management of project; for example; a separate project management unit or appointment of project management expert. In particular, it could be argued that effective project management challenges the ways in which the organisation works and functions. Instead of simply conceptualising the task demands generated by the project as issues that can be dealt with by a normal (Perrow, 1984) ways of working, organisations need to ensure that they identify and develop the dynamic capabilities (Ambrosini, Bowman, & Collier, 2009; Easterby-Smith, Lyles, & Peteraf, 2009; Macher & Mowery, 2009) that are needed to cope with the task demands of the project.

1.2. Focus of the research

The focus of the research concerns the development and implementation of IT software project which involved many risks that could contribute the failures of the software project. With the complexity of the software project and the risk factors involved, further research in this area deemed necessary as explained further in later paragraphs.

First, research on IT software implementation projects indicates that most projects fail to deliver business value in the way anticipated or they do not function at all (Kumar, 2002; Procaccino *et al.* 2005; Rubinstein 2007; Sauer *et al.* 2007). These failures represent significant service delivery problems that, in modern 'tightly coupled' and 'interactively complex' organisations (Perrow, 1984) can generate quite fundamental problems around organisational strategy and performance. For example, Ewusi-Mensah & Przasnyski (1991) report that 35% of failing or troubled software projects were abandoned *after* the implementation stage; the Keil & Mann (1997) study of IS auditors found that about 30-40% of software projects showed some degree of project escalation; and industry surveys suggest that only about 25% of software projects succeed outright (that is, completed as scheduled, budgeted and specified (Bannermann, 2008)). The cost of this level of failure is significant (Charette, 2005), global (KPMG, 2005), and is widespread across the economy (Sauer & Cuthbertson, 2003).

Second, there are issues relating to the 'boundaries of consideration' (Jackson, 2002) that managers impose on projects and the implications that these may have in terms of constraining the processes around decision making and planning. These issues are at the core of management functions and processes and generate challenges in terms of the predictive validity associated with decision making and the evidence-base on which those decisions are made. Seyedhoseini, et al

(2009) argue that time quality and cost are often seen as the central elements of a successful project. A selective focus on a narrow set of issues without giving due consideration to the wider factors that can influence the success of a project is inherently problematic.

Third, projects provide important insights into the ways that emergent properties impact upon the control and cohesion of strategy. Of particular importance here is the manner in which failures around cohesion and control can generate emergent risks for the organisation. The problematic nature of the risks associated with IT software projects is not a new area of research (Block 1983), nor is the fact that proper management of such risks affects the success or failure of a project (Han & Huang, 2007; Jiang & Klein, 2000), but the perception of project risk and creation of an appropriate risk construct remains an area of significant contention. It is this, the aim is to address and explore, and to assess the implications it has for the generation and management of risk as it relates to management of IT software projects.

1.3. Risk construct of IT software project

An important element of the debate relates to a set of unresolved issues surrounding the failure to understand, identify and manage these risks, and which are often cited as contributing factors in software project failures (Keil *et al.* 2000). Specifically, the research focus, concerns the absence of an agreed risk construct for IT software projects over which there has been academic disagreement for some years (Wallace & Keil, 2004; Barki et al, 1993). The key issue appears to be the lack of a systematic method by which the risk construct is developed that enabled initial problem boundaries to be identified. This has led to a variety in the number and descriptions used for candidate risk factors. The consequence is that, there has been:

> *"little attempt to move beyond checklists to establish the underlying dimensions of the software project risk construct or to develop good instruments for assessing software project risk." (Wallace & Keil, 2004, pg 291)."*

Additionally, as reported in Keil et al (2008), there is somewhat mixed evidence of the use of risk checklists: the study finds more risks being identified but that this had no impact on decision-making and also, importantly for this research, that the checklists help shape risk perception. This has its own risks, as the authors point out

> *"To the extent that the risk checklist is not comprehensive or is biased toward certain risks and away from others, this can impact risk perception, as the checklist serves as an attention-shaping mechanism." (Keil et al 2008, Page 915)*

The proposal within this research is to address this aspect of the debate by seeking to clarify the role of a project life cycle as a frame of reference that contracting parties might agree upon and which should act as the basis for the project risk construct. The focus on the project life cycle as a risk frame of reference potentially leads to a common, practical view of the (multi) dimensionality setting of risk within which risk factors may be identified and which believe to be grounded across a wide range of projects and, specifically in this research, for IT software projects. A key dynamic here is the manner in which organisations embrace the project as a key component of normal management approaches to risk – thereby increasing the sense of cohesion and control around the process – rather than seeing it as an external issue to be managed by 'experts'. The incorporation of the project allows for a challenge to be mounted to the core assumptions and beliefs of both the organisation's core management as well as its project teams.

1.4. Aims & Objectives

The purpose and contribution of this research is underpinned by an exploration of the nature of a risk construct as it relates to IT software project failure within the project life cycle and to propose a viable method at addressing this, as yet, unresolved issue. This will be able to coherently address the key aims of the research, which are :-

- Ascertaining what risk factors are most likely to emerge (the 'risk likelihood' question);
- To identify the impact of the emerging risk factors in relation to cost overrun (the 'risk impact' question);
- To reveal which risk mitigation strategies are likely to be used in practice (the 'risk resolution' question).
- The mitigation strategies will be empirically correlated to both the most likely risks and those most important to cost overrun so as to link, in a manner not before reported, how risk is addressed in IT software projects.
- Through the rating, ranking, and common groupings or clustering of the risk factors, fuzzy models will also be developed in assisting the practitioners assessing the risk factors within the life cycle stages.

The research seeks to address directly how organisations can ensure that they identify and develop the dynamic managerial capabilities that are needed to cope with the task demands of the risks inherent in the implementation of IT software projects. The research believe that this approach is theoretically coherent and represents a consistent explanation of risk management used broadly in practice.

1.5. Problem Statement

The manner in which risk is identified, quantified and decisions taken about its consequences forms part of an important element in the development of an organisation's strategic management process and has a particular salience within project management. The method by which 'risk' is organisationally 'constructed' implies a pre-conception of the boundaries of consideration that serves to shape the organisation's portfolio of risks and from which the methods of managing those risks is derived.

The fact that there is a degree of 'agreement' about a risk and its expression does not, of course, mean that a risk construct has predictive validity. An agreed risk 'construct' is a starting point for dealing with risk within project management and it is a dynamic expression that is subject to change. The problem is essentially one of construct validity in which risk factors are understood, identified and communicated effectively. Wallace *et al.* (2004) argue that the linkages between IT software project risk and performance are often missing and that risk checklists and existing risk frameworks do not connect to project performance. The inevitable consequence will be the failure to create relevant risk mitigation strategies as the assumptions and core beliefs of decision makers will shape the 'lens' through which the problem is viewed.

Risk within organisations is invariably grounded within an analytical framework and a data set that can have varying degrees of robustness. It is not, and perhaps never can be, a precise calculus given the inherent ambiguities that surround major projects. The demarcation of IT projects into project life cycle stages responds directly to this complexity and to calls for research in the area (Ewusi-Mensah & Przasnyski, 1994). Work by Ewusi-Mensah & Przasnyski (1994), for example, looked at project abandonment and called for further work in relation to project stages. As a consequence, the complex cause-effect-response relationships in risk management of IT projects remains an inadequately addressed problem. Moreover, a key dynamic of this process relates to the ways in which project teams are constructed, resourced and, in some cases, outsourced.

The use of outside consultants in software projects is extensive (Sumner, 2000) and, with projects that engage outside professionals to assist, it would seem essential that agreement between user and outside service provider (the 'IT professional') on all aspects of the project is in place before work begins and resources are committed. The agreement between the contracting parties then becomes the frame (or reference point) for project process, delivery and cost. It is within this context that risk factors could be specifically identified, understood, addressed and risk consequences managed. The frame of reference that the contract represents then emerges as an agreed risk construct based on both the organisational context of the client and on the project experience of the IT professional. What the research seek to illustrate is that agreement concerning risk constructs between contracting parties (in this case the IT user and IT professional) are freely entered into and reflect a common understanding of IT software project risk exposure.

The research priori view is that the perspective of IT professionals is an important one to obtain (Keil et al 2008) but is not one which, in this study, would necessarily produce a bias in the results arising from lack of scope of perspective. However, it need to point-out that the risk construct is not exclusively drawn-up with IT professionals in mind and which would therefore limit the generality of the results. The construct is general, the responses, of course, might vary dependent on which stakeholder is asked. The conclusions drawn therefore relate to the opinions of IT professionals and hence the scope of the study in terms of stakeholders is limited but which the research argue has resonance to the views of other stakeholders if the risk construct is accepted and evidenced as a valid construct. The methodology is premised on an agreed risk construct and, it is within that agreed context, that different views are sought. It would seem both inappropriate and potentially damaging to offer a different risk construct when a framework that is likely to represent the foundation of a risk construct already exists.

Such an approach leads to an immediate and obvious interpretability of risk factors which has been a problem previously encountered (McFarlan, 1981; Barki *et al.* 1993; and Jiang & Klein, 2001). Moreover, it is distinguish in this survey and analysis between the *identification* and *impact* of risk factors as related to IT software project failure. That is, the research provides evidence on IT software project risk from both identification and consequentiality perspectives. The research impact measure is referenced to project cost overruns. This is not a comprehensive measure but does provide at least one metric which is firmly placed into a wider risk construct, which go on to elaborate, and which directly addresses the reports of inadequately few attempts to *measure* IT software project risk (Wallace *et al.* 2004; Barki, *et al.* 1993; Ewusi-Mensah (1997): and Yetton, *et al.* 2000).

Whilst there are many different, broad and overlapping definitions of project success and failure (Baccarini, 2004; Linberg, 1999; Proccacino *et al.* 2005; Ropponen & Lyytinen, 2000;), completing the project within the estimated budget remains the project characterisation that is most frequently mentioned with respect to the reporting of project failure (see Atkinson, 1999). Whichever way 'project failure' is characterised, it is hard to escape the idea of there being either a direct or indirect project cost implication (Mata *et al.*,1995, Reel, 1999, and Melville, *et al.*, 2004;).

Lopes & Flavell, (1998) explore in a survey study a broad number of dimensions of project failure and argue that concentrating on cost and technical failure is too narrow a definition of 'failure'. They do not dismiss the importance of costs and emphasise that cost/profit considerations cannot be 'subjugated' since the financial viability of the private firm supports its principal objective: as they put it, 'survival'. Notwithstanding the importance of financial factors, the authors go on to provide examples of non-financial factors that should be considered (such as broader political, environmental, social, and wider organisational aspects). These are wide ranging and both detailed and contextual in that many of the factors refer to the environment within which projects are developed and thus reflect the importance of layers of complexity in projects. This is reinforced by this research earlier comment that projects are 'multi-level issues' (Dansereau & Yammarino. 2002) and, as such, may well be immune to quantitative analysis that accounts for all influences. However, this research focus is for empirical validation of risk likelihood and risk impact as measured by cost overrun within a fairly tightly defined risk construct. This study is consequently limited compared to the factors listed in Lopes and Flavell (1998). The requirement for empirical validation will necessarily precludes considerations of social impact, for example, since they present difficulties for empirical validation.

Lopes & Flavell's conclusion that cost cannot be subjugated is supported in Sumner (2000), for example, who reports that, out of 7 case studies researching risk relating to enterprise-wide system development, 6 of these had cost reduction as a project goal. Furthermore, a cost consequence approach as this research would argue is most readily interpretable by survey participants and is one which naturally accords with a major characterisation of project failure in popular perception. More specifically, the research contention is that a cost focus leads to survey responses that are directly relatable to the experience of the survey participant base (as one party to a cost/fee-based contract), is more likely to be relevant to IT software project risk goals in a commonly understood manner and which, consequently, is likely to enhance the psychometric properties of the research survey results.

1.6. Significance of the study

It is fairly widely reported that the management of risk reduces project failure and improves software project outcomes (for example, Boehm, 1991; Ropponen and Lyytinnen, 2000; and Keil et al, 2008). This is underlined by a set of principles and practices aimed at identifying, analysing and handling risk factors to improve the chances of achieving a successful project outcome or avoid project failure (Boehm 1989, 1991; Charette 1989; Kerzner 2003; Huang and Han, 2008). However, the linkage between identification, quantification and risk mitigation remains largely unexplored in the literature from an empirical perspective. Researchers are mostly unaware of what happens in practice concerning the linkages of different risk stages over a range of IT projects although there have been repeated calls for and references to the need for this evidence (Fairley, 1994; Heemstra and Kusters, 1996; Conrow and Shishido, 1997; Brandon, 2005; and Bannerman, 2008).

Of the work that is available, Tesch et al (2007) identifies avoidance and mitigation strategies that include: a consistent commitment from top management (also identified in Cule at al, 2000), planning and scheduling of project into phases, adequate user involvement, the establishment of good communication lines, and adequate resource requirement planning. The study also suggested that the project team have a written project charter which contains information such as a clear goals and requirements, clearly outlined deliverables and success criteria, contingency or back up plans, and a roles and responsibilities matrix. Mahaney & Lederer (2009) focus on the impact of monitoring and regular updating of the project as important strategies. Their study revealed that thorough monitoring and regular updating could provide information that the project is progressing within budget, on schedule and achieving quality expectations. Shih-Chieh Su et al (2008) highlighted the users' perspective to control progress and act as product quality gate-keepers in the software development process.

What seems apparent from the extant research relates to the twin deficiencies of a lack of agreed risk construct combined with an empirical validation of the links between risk likelihood, risk impact, and evidence of strategic response in terms of risk mitigation. The risk construct the research propose and the evidence it seek to accumulate is designed to address these deficiencies. The problem boundary is set by the proposed risk construct within the context of the project life cycle. The survey evidence then directly reports to key areas of risk identification, quantification and mitigation. The mitigation strategies reported are empirically those most likely to be observed as addressing the most likely risks and those most closely associated with cost overrun.

1.7. Methodology outline

The research methodology is generally outlined in this paragraph. However, the subsequently detail of the research methodology approach with further argument and empirical support, is elaborated in the later methodology chapter.

The research reports an analysis of a survey of IT professionals with the objective of ascertaining their views concerning the identification, impact evaluation, and possible mitigation of the risks involved in supporting the development of IT software projects. The research method comprises a number of tasks and phases. The early phase involves the literature reviews on the subject area to gain more insight into the risk factors that may affect a software project. This approach is consistent with how other researchers conducted their research in this area. Eun Hee Kim et al (2006) and Perera et al (2006) also used extensive literature reviews of IT project risk factors, and produce a risk list for their study. Even Wat & Ngai (2005), Wallace et al (2004), Ropponen & Lyytinen (2000) used the same method. This resulted in a list of overlapping risk factors, as other researchers also tends to use previous researchers list for their own research, with the additional of some new risk factors. Apart from the literature reviews, other risk factors were also identified through the researcher's knowledge, understanding, informal discussion and brainstorming sessions with others.

The next step involves identifying an appropriate risk construct based on the project management perspectives of project life cycle and then to produce a list of candidate risk factors to be included in the survey. The list of risk candidates were categorised and interpreted in terms of the risk construct. The major empirical analysis is undertaken in which determining how survey participants ranked IT software project risk likelihood of emergence and impact of the emerging risks on cost overruns. The survey respondents were also asked to rank the effectiveness of risk mitigation strategies from a comprehensive list drawn from the literature. The ranked risk mitigation strategies were then correlated to the extracted risk likelihood and to cost overrun components.

The questionnaire survey was circulated electronically to software development companies, IT consultancy and management companies and web development companies in the UK, USA, Europe, India, China, Japan, Canada, Australia and some other Asian countries. Their contact and email addresses were obtained from the internet. Respondents were approached across a variety of IT job functions, management hierarchy and with varying amount of experience. A total of 1000 IT/IS related companies with an email address for correspondence was selected as respondents, with 324 returned completed questionnaires.

The completed questionnaire were transferred into a proper tabling and formatting using Microsoft Excel and Statistical Package for Social Science (SPSS) in order to extract relevant statistical data for the research. The statistical findings of the survey, the overall conclusions and implications of the research were presented in the later chapters.

1.8. Outline chapters

Chapter 1 Provides an introduction, purposes and objectives of the research. A theoretical literature review and justification of the significance and importance of the research were outlined. The focus of the research, the research background and the problem statement were elaborated in this chapter.

Chapter 2 The research methodology for the research was highlighted. The approach for the development of the questionnaire design for the survey was elaborated. The research framework and the risk construct proposed for this research were explained. The data collection processed and the analytical methods used for this research were highlighted.

Chapter 3 This chapter highlighted a few of the most common and generic software development life cycles that being used in the software industries. The risks associated with the existing software development life cycle were highlighted. The relevancy of the life cycle from the perspectives of project management was also identified.

Chapter 4 In this chapter, the focus is on extracting the generic risk factors from the literature reviews of software project risk which are common to most projects. The risk factors extracted were organized into the relevant stages of the project management perspective to make them more useful and meaningful for the managers and practitioners. A new framework within the perspectives of the project management life cycle was proposed for the research.

Chapter 5 Data collection was conducted and the descriptive analysis of the responses was performed using the Microsoft Excel and SPSS. The data obtained were grouped and organised in order to be analysed and discussed further in later chapters. The general and overall findings of the survey were presented in this chapter, but the detail discussions of the findings were presented in later chapters. The statistical techniques includes ranking, weighted mean, standard deviation, coefficient of variation and severity index.

Chapter 6 The results of the descriptive analysis from the previous chapter were elaborated and the perceptions of the software practitioners were discussed in details. The main areas of agreements and disagreements of the perceptions among the practitioners were discussed. ANOVA analysis and Post Hoc test also conducted in supporting the arguments.

Chapter 7 Factor analysis processes of the responses was conducted. Using the principal component analysis (PCA) techniques with the SPSS, components of the risk factors were identified based on the factor loading scores. Clustering of the components were established in finding common themes based on the proposed risk construct of project management life cycle.

Chapter 8 The result of the responses regarding the risk management strategies was discussed. The differences in the strategies chosen for different risk components for the risk occurrence and its impact on cost overrun were highlighted among the different practitioners.

Chapter 9 Results from the factor analysis process were used for the modelling using the fuzzy analysis theory. The existing software project risk model was explained and the use of fuzzy modelling techniques was also detailed. The methods and computation of the fuzzy membership function were explained. Risk profiles based on the fuzzy was developed and a hypothetical example of the fuzzy model application was highlighted.

Chapter 10 Discussion of the results of the whole analysis and research conclusion. This chapter discusses the significance and the relevancy of the research within the overall perspective of the software development project. Relationship and correlations of the risk occurrence, its impact on the cost overrun and risk mitigation strategies were identified and explained.

Chapter 11 Summary, Research contribution and recommendations for further research. The whole perspective of the research and the research contribution towards the practical and application perspectives were highlighted. The limitations of the research and the recommendations for further research were also explained.

CHAPTER 2 RESEARCH METHODOLOGY

2.1. Introduction

The purpose of this chapter is to present the methodologies and processes employed to develop and design the questionnaire for the research. Even though there were some advantages and disadvantages of a survey method (Bloom & Fisher, 1982; Raj, 1972; Burns, 2000, Newell, 1993), but most researchers agreed that a survey method is the effective and preferred methods for statistical data and opinion collection. Also, through the literature reviews, most of the researchers in this area were conducted through a survey, rather than other methods such as interviews or case study (Keil et al, 2008; Costa et al, 2007; Wallace at al, 2004, Schmidt, 2001; Ropponen, 2000; Jiang et al 1999,2000).

A questionnaire survey was conducted for this research in order to identify the likelihood of risk factors and its impact on cost overrun of a software project. The data obtained through the questionnaire was analysed using the Microsoft Excel and SPSS to assess the perceptions of the practitioners/respondents. These data were also used to develop a fuzzy modeling of the likelihood of risk factors and its impact on cost overrun.

The methodology is outlined in Figure 2.1 and the approach is subsequently detailed with further argument and empirical support.

2.2. Research method

The research method comprises a number of tasks as shown in Figure 2.2. *Task 1* involves identifying an appropriate risk construct and then to produce a list of candidate risk factors to be included in the survey. The list of risk factors extracted from the literature reviews was incorporated in the risk construct in terms of project management life cycle stages. The questionnaire design was developed to be used in the data collection phase. The data collection is undertaken as *Task 2* and the statistically validity of the sample also detailed. The analytical choice is explained in *Task 3* which is designed to discover, from the survey evidence, how important risks may be categorised and then interpreted in terms of our risk construct. The descriptive statistics of the survey were analysed within this task.

Figure 2.1 : The Research Methodology

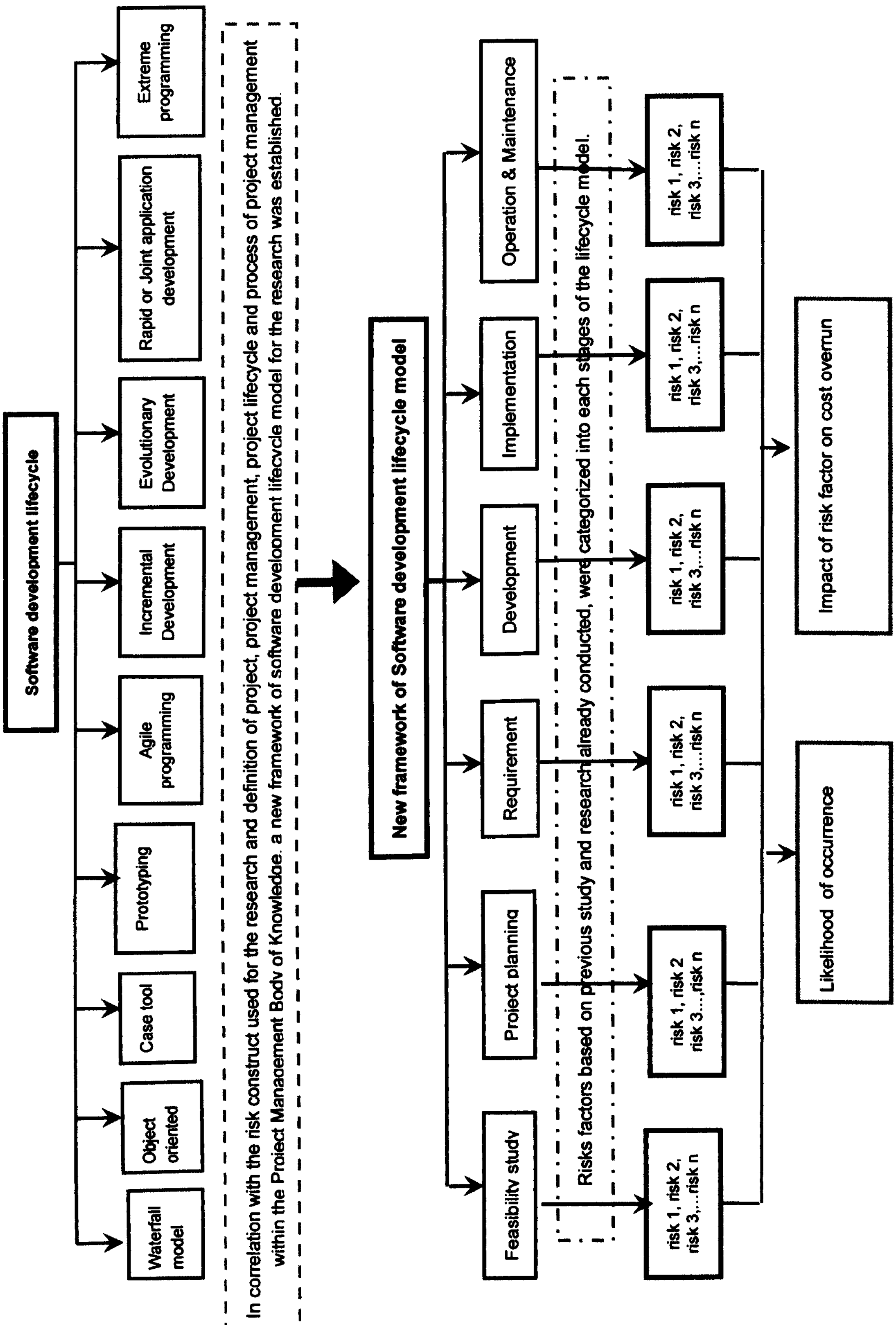
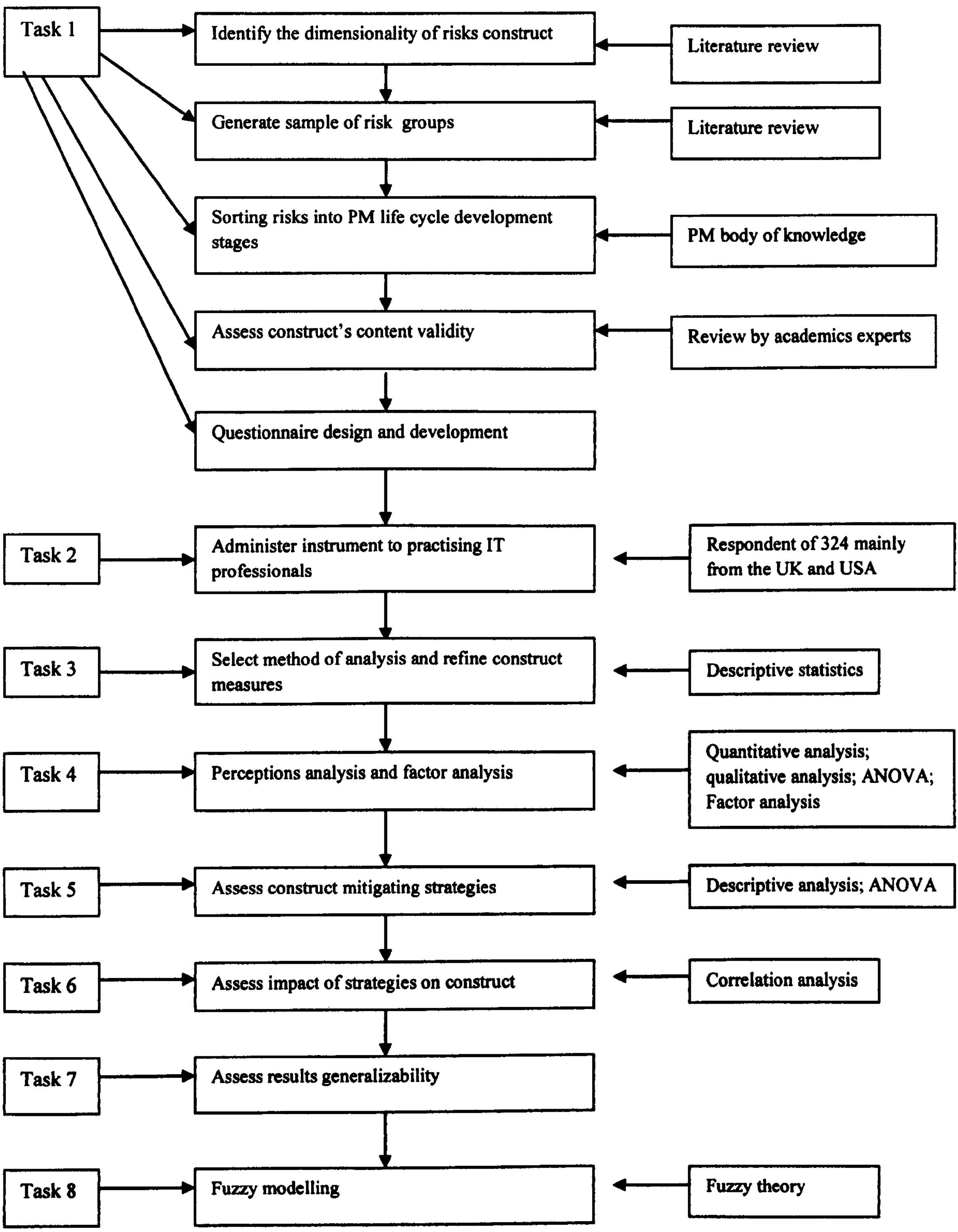


Figure 2.2 : The process used in the development, validation and interpretation of the instrument
Source: adapted from Smith et al. (1996), and Wallace, et al. (2004)



The major empirical analysis is undertaken in *Task 4* in which determine how survey participants ranked IT software project risk likelihood and cost overruns. The perceptions of the software practitioners were differentiated for the risk likelihood and its impact on cost overruns. The research hypothesis used for this task (based on 5% significance level) were :-

H_0 *($p > 0.05$) - There is no significant different among the respondents rating for the likelihood occurrence of risk factors*

H_1 *($p < 0.05$) - At least one of the groups rating for the risk factors likelihood occurrence significantly different from at least one other groups*

The 5% significance level was also used for the research hypothesis of the risk factors impact on cost overrun :-

H_0 *($p > 0.05$) - There is no significant different among the respondents rating for the risk factors impact on the cost overrun of software project.*

H_1 *($p < 0.05$) - At least one of the groups rating for the risk factors impact on the cost overrun significantly different from at least one other groups*

The survey respondents were also asked to rank risk mitigation strategies from a comprehensive list drawn from the literature. Thus, *Task 5* then determines which risk mitigation strategies were most important according to the survey. This was undertaken both for risk likelihood and the risk impact on cost overruns. The effectiveness of the risk mitigation strategies were looked upon in relation to the reduction of IT software development risks, without specifying their relationship to risk likelihood or the risk impact on cost overrun. The research hypothesis used for this task was :-

H_0 *($p > 0.05$): There is no significant difference among the respondents rating for the effectiveness of the risk mitigation strategies.*

H_1 *($p < 0.05$) : At least one of the groups rating for the effectiveness of the risk mitigation strategies significantly different from at least one other groups.*

Thus, *Task 6* correlates mitigation strategies to risk components, for both likelihood and cost overrun models that are extracted from factor analysis to determine which risk mitigation strategies are effective in practice. Overall conclusions and implications of the research are drawn in *Task 7*. Finally, in *Task 8*, a model of fuzzy risk analysis was proposed to assist software IT practitioners and decision makers in formalising subjective thinking that are required in assessing the current risk environment of their software development in a more systematic manner.

2.3. Risk construct of the research

The survey directly asks participants their view about the core research questions concerning risk identification, quantification or impact, and mitigation: that is, detailed questions that elicit opinion on:

i. the most likely risks to emerge (the 'risk likelihood' question);

ii. the impact of the most likely risks to emerge in terms cost overrun (the risk impact question); and

iii. the most effective mitigation strategies that reduce the risks (the 'risk resolution' question).

The questions frame in detail from the research candidate risk factors and candidate mitigation strategies that the research survey opinion on. Data was drawn from respondents who were asked to rate risk factors in terms of its likelihood occurrence using a 5 point Likert scale (none; unlikely; likely; highly likely; very highly likely).

Respondents were also asked to rate each risk factor impact on the cost overrun of a software project using a 5 point Likert scale (Very low(1-10% overrun); Low (11-20% overrun); Moderate (21-30% overrun); High (31-40% overrun); Very High (more than 40% overrun). For risk mitigation strategies, a 7 point Likert scale is employed (don't know; not effective; very slightly effective; generally effective; highly effective; very highly effective; exceptionally effective). The method of framing the survey questions is determined by the risk construct employed for the research.

Thus, in *Task 1,* the research starting point is to derive the proposed risk construct from the project management models available for software development project. The available models are themselves descriptions, in one form or another, of stereotypical whole project life cycles. In that sense, represent putative risk constructs, since they articulate project activities that require active management, including risk management from which were developed for the research. There are a range of project management models used in practice for software development project (for example the Waterfall Model (Alter, 1992), the Evolutionary and Incremental Development model (Larman & Basili, 2003), Object Oriented models (Henderson-Sellers & Edwards, 1990), Agile programming, Prototyping and CASE tools. Inevitably, there is a lack of consensus on how to view or model the software development life cycle (Alter, 2001) which has resulted in non-similarities of stage descriptions in survey responses in previous work (Davis *et al.* 1998). This, of itself, poses a problem for risk construct development and the manner in which risk is conceptualised which the research addressing.

The available literature has produced a number of conceptual frameworks to explain different types of software development risk, risk management strategies, and measures of software project performance (Wallace & Keil, 2004; Na et al, 2007). In an investigation-based literature review, Susan and Alter (2004a & 2004b) address the limitations and shortcomings of the IT risk classification models. Their work used the risk lists produced by previous researchers (Boehm 1991, Barki et al 1993, Keil et al 1998, Jiang & Klien 1999, 2000; Ropponen and Lyytinen 2000; Schmidt et al 2000, Keil et al 2002, Addison 2003, Wallace et al 2004, and Wallace & Keil 2004). The study concluded that existing IT risk classification models have limited applicability and lack ability to easily communicate an organising framework for IT risk factors: that is, a risk construct. In reinforcing this, the investigation also highlighted that IT risk classification models should be based on clearly defined concepts that are understandable, adaptable to a variety of contexts, and of practical use.

The basis in developing this line of enquiry is that categories, groupings and dimensions of risk factors based on project management perspectives could potentially provide a broader framing – and hence be more widely applicable - for thinking about what risks might be targeted for risk mitigation. Such categorisations could be empirically derived from existing project management practices and would therefore arguably be closer to management practice. Wallace et al (2004) and Ropponen and Lyytinen (1996) stress the importance of empirically deriving the sources and types of risks associated with software development projects. A further motivation for employing a project management perspective is that resulting risk categorisations will be placed in context of the stages, processes and activities within the software development project. As such, risk management becomes engaged with project management in a clearer manner.

Kuruppuarachchi et al (2002) stress that, although many factors are involved to achieve IT project success, having a detailed action with identifiable stages in a project lifecycle context could increase the chances of success. Meredith & Mantel (2007) also highlighted that organising risk factors in a project management framework facilitates a focus on roles and responsibilities and allows for the coordination and integration of activities for regular monitoring and aligning with the project goals. Such a taxonomy would then, arguably, better enable management to identify and manage risk as they emerge with project stages and more closely reflect project activity, allow more accurate assessment of the level of impact, and facilitate the identification of the most appropriate risk mitigation strategies.

2.4. Risk construct framework

The research advancements in this area are to seek to harmonise the classification of risk factors according to the life cycle stages by making reference, initially, to the Project Management Body of Knowledge. (PMI, 2004).

This encompasses definitions of "project", "project management", "project life cycle" and the process of project management . From this, the research produce a specification of the software development life cycle into six broad stages which forms the structure of the questionnaire. The stages are Feasibility study (FS), Project planning (PP), Requirements (R), Development (D), Implementation (IM), and Operation and maintenance (OP). The candidate risk factors are extracted from a review of the literature of the risks involved during the software development life cycle and these amounted to 104 individually identified risk factors. The risk factors then mapped into one of the 6 stages of the software development life cycle which allows the research to categorise the questionnaire in terms of the risk construct/project life cycle adopted. This approach adopts most of the instrument validation phases employed in Wallace et al (2004) which describe in some detail. Their first phase of identifying an initial risk construct was sourced ('culled', in their language, pg 296) from a review of the literature. The research approach has the same phase but differs in that this research employed project management stages using PMI which is itself a synthesis of project management approaches. In so doing the research seeks to employ an initial risk construct which, from the outset, accords with existing project management practice.

In their second phase, Wallace et al (2004) then go onto generate candidate risk factors and, because they were initially generated by the researchers, they then subject the candidate risk factors to a variety of sorting of pre-testing procedures to validate the words used to describe the candidate risk factors and their interpretability as attributes of the risk construct they attempt to investigate. This is unnecessary for this research candidate risk factors since they are sourced from the literature, hence do not face the problem of validating self-generated candidate risk factors. This does not imply a flaw in Wallace and her co-author's approach since they are rigorous in testing what they have self-generated.

In phase 3, Wallace et al (2004) pre-test their survey instrument with software project managers attending a conference. This research pre-tests the questionnaire by asking experienced academics and researchers as an initial pre-survey screen. These experienced academics consist of those who had already published their researches in this area, and their research outputs were also used as part of the literature reviews. Responses were received from Verner (2002, 2005, 2007), Klein, (2000,2001,2002), Kwan (2004, 2007) and Barros (2004, 2007). Their comments and suggestions were considered and amendments were made to the questionnaire.

The risk factors were also subjected to post-survey validation by testing if it is possible to categorise them into broader risk themes (that is, the risk construct). This is undertaken using Bartlett's Test of Sphericity and the results reported below in Table 2.1.

Bartlett's test only examines if the correlation matrix between the risk factors is an identity matrix (the null hypothesis_ and hence that is possible if the null is rejected to assert - to a probability – that the candidate risk factors may be categorised. The fourth phase of the Wallace et al (2004) research is to conduct the survey leading to a sample of 507 (response rate of 13.3%) at which point this research approaches coincide.

Stages	Bartlett's Test of Sphericity (Significance value)	
	Likelihood occurrence	Impact on cost overrun
Feasibility study	0.000	0.000
Project planning	0.000	0.000
Requirement	0.000	0.000
Development	0.000	0.000
Implementation	0.000	0.000
Operation maintenance	0.000	0.000

Table 2.1 : Bartlett Test of Sphericity shows the significant value of $p < 0.05$, which means the correlation is not identity matrix; i.e, it is possible to categorise them into broader risk themes (that is, the risk construct)

Bartlett's measure test the null hypothesis ($H_0 > 0.05$) that the original correlation matrix is an identity matrix. For the possibilities of groupings or categorizing of the risk factors to work, it needs some relationships between variables that measure similar things or have common trend and themes. Using a significance level of 0.05, the Bartlett's test shows the values of p for both likelihood occurrence and impact on cost overrun are highly significant. This test tells us that the correlation matrix is not an identity matrix; therefore, there are some relationships between the variables. The results of the Bartlett test proved that categorizing these variables into broader risk themes are possible.

2.5. Questionnaire design

Many researchers believed that a questionnaire is the fastest and the most effective technique for statistical data and opinion collection (Parasumraman, 1991; Burns, 2000; Bloom & FRisher, 1982), compared to the other methods. Questionnaires methods provides enough time for respondents to consider their responses carefully without interference from an interviewer or caller for instance. Questionnaires are less intrusive than other approaches, are familiar techniques of obtaining information and data to most people and hence do not make them apprehensive. Questionnaires can address many issues and questions of concern and can be employed for studies involved in a high number of sample size or vast geographical region. As this study involved high number of risk factors and required a large sample size, questionnaires deemed to be the appropriate option. The list of risk factors were collected and sourced from the relevant literatures in this area, with the addition of risk factors identified based on researchers knowledge, understanding and discussions with supervisors and fellow researchers.

The draft questionnaire produced was shown to supervisors, colleagues, fellow researchers and requesting comments or suggestions in terms of wordings, sequence of questions, layout of the questionnaire and its contents. The suggestions include rewording the questions; review the sequences of the risk factors and also questionnaire design layout. The review of the draft questionnaire was done a few times to establish clearer questions, wordings, layout in the design of the questionnaire. Wallace et al (2004), pre-test their survey instrument with software project managers attending conference. This research pre-tests the questionnaire through a pilot study by sending the questionnaires to experience academics and researchers in this area. These experience researchers includes those who had already published their researches in this area for the past years. Their research outputs were mainly used for references by other researchers and this research. Their comments and suggestions were considered and amendments were made to the questionnaire. This can ensure that the questions were comprehensible and most likely to produce useful data.

Questionnaire design requires many modifications, editions and corrections before a final draft creation ready for data collection. The main objective was to produce a final questionnaire which makes the process of responding easier, more fluent and clearer for participants. In addition, this may resulted in a good response rate. For this research, respondents are requested to rate each risk factor using the 'Tick boxes' or rating approach, on the basis of their experience and opinions to a certain statement, by asking them to rate its significance on a rating scale or Likert scale. For this research, the scale range is from 0-5.

The design of the questionnaire was consistent with some other research conducted in this area of study. (Jiang & Klien ,1999; Ropponen & Lyytinen,2000; Cule et al,2000; Schmidt et al, 2001; Wat & Ngai, 2005; Procaccino et al, 2005; Perera et al,2006; and Eun Hee Kim,2006). However, they were differences among the previous rating method, in terms of the range of scales being used.

The diagrammatic representation of the process flow of the questionnaire design was illustrated in Figure 2.3.

2.6. Questionnaire structure

The main questionnaire survey consists of three parts; Part 1, addresses the general information about the respondents background and experience; Part 2, lists the selected risk factors and the respondents need to assess and rate the degree of significance of each risk factor using the 5-point scale. Part 3 is about asking the opinions of respondents in relation to the effective risk mitigation strategies.

In Part 1 of the questionnaire, general information about the respondents were asked, like their background, nature of businesses, experience in years and number of projects involved. For this research, three main category of businesses related to IT/IS were chosen; Software development companies; IT consultancy & management companies and Web development companies. Although, software project is more related to software development companies, but due to the diversity of services that other IT/IS related companies might offer to their client than their name suggest, the research need to include a more general IT/IS related companies as well. This three main category of businesses should cover the majority of the population, if not all. Additional information was also requested from the respondents, regarding their risk management criteria, such as their risk management expertise and their risk management practice with their organization.

From previous research, most of the studies were conducted by interviewing, role playing, case studies or doing a survey on the software projects within the perceptions of the Project managers or Users. In fact, the previous research samples were predominantly Project managers. (Costa et al, 2007; Tesch et al, 2007; Wallace & Keil, 2004; Wallace et al, 2004; Schmidt et al, 2001; Ropponen & Lyytinnen, 2000; Jiang & Klien, 1999,2000; Keil et al, 1998; Barki et al, 1993; Boehm, 1991).

For this research, the focus of the respondents will be within the development team itself, which include Project managers, Software Developers/programmers, IT technical support staff, and also the Managing Directors/Board of Directors. The Managing Directors/ Board of Directors perceptions was perceived to be relevant as the risk factors within the project could have very significant effect on the running of the business as a whole. Whereas the IT technical support perceptions also deemed to be very important as they are probably going to have a direct involvement with the users during implementation and providing technical support when the software is up and running. Users perspectives was not put through, as it is being done before by Keil et al (2002) where his research found differences between project managers and users in the risk factors they identified and their relative importance in software projects. Addison (2003) also studied the differences of opinions of the most important risks factors in the development of E-commerce project between project managers and users, using the Delphi technique.

Research also shows that project managers groups and users tend to identify and rank highly risks that are perceived to be outside their own control (Bannerman, 2008). That is, they tend to identify risks in the responsibility domains of others, rather than point to factors as risks within their own areas of responsibility (March and Shapira, 1987; Schmidt et al., 2001). It is crucial that the views of all key software practitioners groups are taken into account in the risk identification and management process.

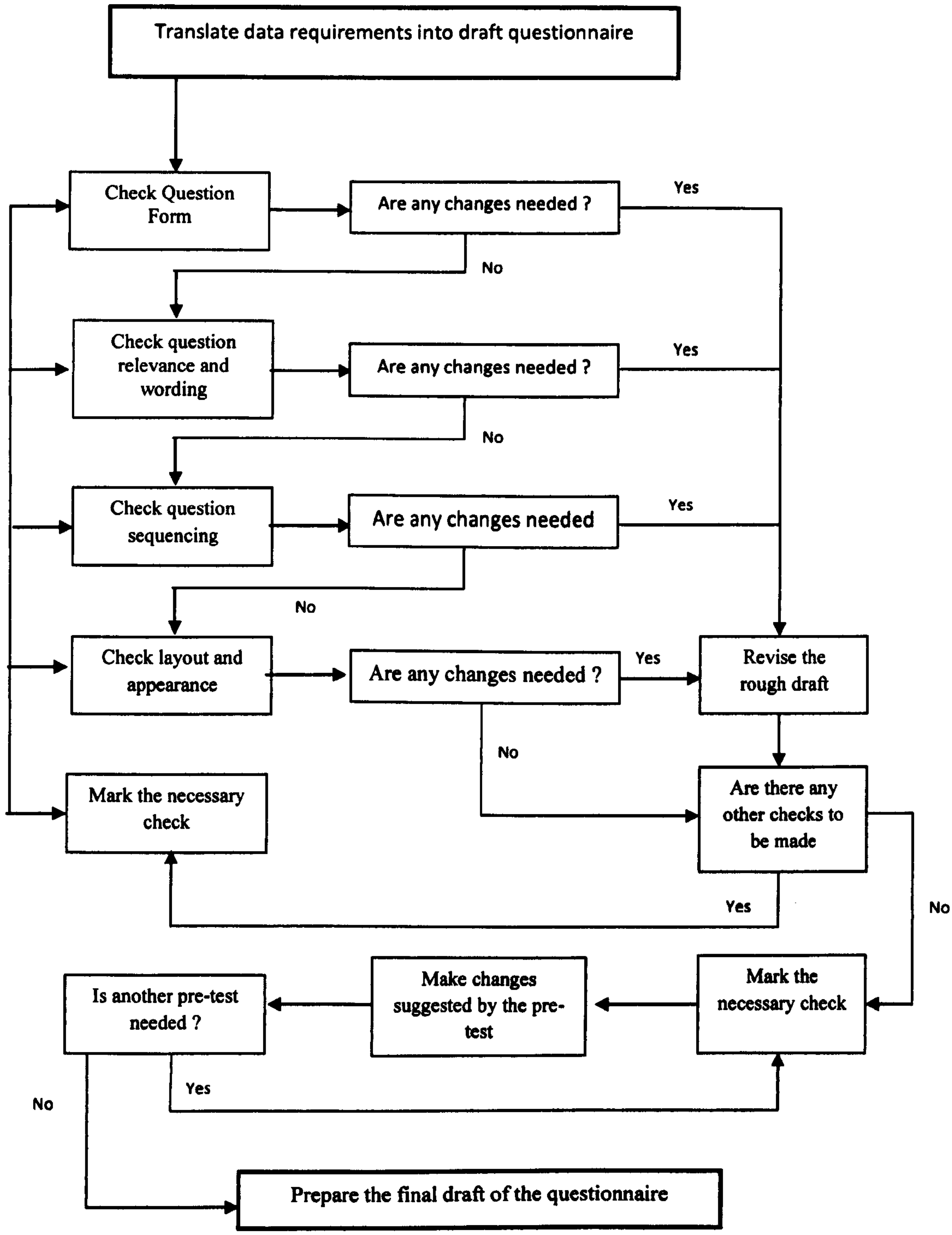


Figure 2.3 : Components of questionnaire design process

In Part II, respondents were provided with a list of risk factors related to software project, divided into 6 stages of development lifecycle based on project management perspective. The respondents need to assess the likelihood occurrence of each risk factors on a scale of 1 to 5, where :-

1-denote; None occurrence;

2-Unlikely;

3-Likely;

4-Highly likely;

5-Very highly likely.

The respondents were also asked to rate the impact of risk factors on cost overrun of the software project as a percentage of the original estimate, also on a scale of 1-5, where :-

1-Very low (1-10% overrun);

2-Low (11-20% overrun);

3-Moderate (21-30% overrun);

4-High (31-40% overrun);

5-Very high (>40% overrun).

In Part III, respondents were asked to rate the effectiveness of the risk strategies in response to the risk factors using the scale of 0-6 as below:-

0 - don't know

1 – not effective strategy

2 – very slightly effective strategy

3 – generally effective strategy

4 – highly effective strategy

5 – very highly effective strategy

6 – exceptionally effective strategy

For this Part III, the research look for perceptions of practitioners on the effectiveness of risk mitigation strategies in reducing the risk generally, without specifying their relationship to the risk likelihood or impact on cost overrun. The relationship exercise is undertaken later as part of the correlation analysis of factors scores in extracted components from risk likelihood and impact on cost overrun to these mitigating strategies.

The 5-point and 7-point scale are commonly used in a questionnaire survey. Limiting the scales would simplify the data coding or transfer of responses for further analysis. The 5-point and 7-point had a mid-value, whereby the respondents can express their views and perceptions in both direction equally. The respondents can also give a neutral value if they don't have very definite choices. The 5-point scale is used for the risk occurrence and impact on cost overrun, as it involve 5 scales for 104 risk factors, which is already produced a high number of matrix. A more detail scale of 7-point scale is used for the effectiveness of risk strategies, as it only involve 30 mitigation strategies.

2.7. Data collection and survey validity

The questionnaire survey was circulated electronically in *Task 2* in Figure 2.2 to software development companies, IT consultancy and management companies and web development companies in the UK, USA, Europe, India, China, Japan, Canada, Australia and some other Asian countries during the early and mid 2008. Their contact and email addresses were obtained from the internet. Respondents were approached across a variety of IT job functions, management hierarchy and with varying amount of experience. A total of 324 valid questionnaires were returned which constitutes a response rate of 32.4%. The research sample size is at a level that is likely to ensure component estimate stability (see Guadagnoli & Velicer, 1988). A summarised profile of respondents is shown in Table 2.2.

Over half of the respondents (61%) had more than 10 years of experience in software project development with an average of 11.8 years (standard deviation of 5.29). Also, more than half of the respondents had been involved with in excess of 6 projects with an average of 9 (standard deviation of 5.31). The research sample base compares favourably with other studies in this area. Chang (2006), for example, uses a sample base of 219 for a quantitative survey in a study of enterprise information system importance, implementation and benefits. Keil *et al.* (2002), using a Delphi approach, based their findings on 15 managers with an average of 9.5 years of experience and a minimum involvement of 12 projects. Liu *et al.* (2009) employ 34 project managers and 30 senior executives in looking at different perspectives on project risk (in another Delphi study. Finally, Wallace at al (2004) report the responses obtained from 507 software project managers concerning their risk constructs. The response *rate* reported in this study is close to that in the work by Ropponen & Lytinen (2000) who reported 83 responses representing 33.5% of the requests for information circulated.

The geographical spread of respondents, whilst from a wider base than nearly every other study is, nonetheless dominated by the USA and UK which accords with previous research samples. Thus, the survey respondents reflect geographical sources of other studies but are extended geographically. No other country contains more than 10% of the survey sample and hence do not expect any geographical skewness in the results that is different from studies focussing on the USA and the UK. Of the studies reported previously that were used to compare samples, the following sample survey geographical characteristics were reported; Chang 2006, not specified, but likely to be USA given the reporting of the results; Keil et al 2002, USA only; Liu et al 2009, China only; and Wallace et al 2004, unspecified. As a consequence, it is felt that the respondent sample reported here is broadly comparable with other research in this area.

Given the dominance of the US and UK in the research sample, it do not seem that the research sample composition has significant different from other researchers in the field and the limited disparity in views not a fundamental undermining of the research study as a basis of comparison with earlier works.

Table 2.2: Background data of respondents experience in software projects

Participant characteristic	Responses	Percentage (%)
Company type		
Software development company	122	37.7
IT consultancy & management	104	32.1
Web development	98	30.2
Years of experience in software projects		
Less than 3	23	7.1
3 - 6	47	14.5
7 - 10	55	17.0
11 - 14	81	25.0
15 - 18	86	26.5
More than 18	32	9.9
Number of software projects		
Less than 3	27	8.3
3 - 6	103	31.8
7 - 10	83	25.6
11 - 14	48	14.8
15 - 18	40	12.4
More than 18	23	7.1
Geographic distribution		
USA	131	40.4
UK	81	25.0
India	28	8.6
Canada	18	5.6
Others	174	20.4

The table reports survey respondent background and experiential details. Participation rate was 32.4% from 1000 requests. Table reports details of participant company background, individual length of experience in software projects, the number of software projects involved with and geographic distribution of survey participants. Individual countries reported for geographic distribution where respondent numbers exceeded 10.

2.8. Analytical methods

The objective in the research analytical choice, in *Task 3, Task 4 and Task 5,* is to employ a method that enables discovery, from the survey evidence, of how important risks may be categorised and then interpreted in terms of the risk construct. Using a Likert scale, participants were asked to rank, for IT projects, from a pre-defined list:

i. the most likely risks to emerge;

ii. the impact (in terms of percentage cost loss) of emerging ; and

iii. the most effective mitigation strategies that reduce risk.

For questions in (i) and (ii) above, the survey responses were analysed using principal components analysis (PCA) to determine which of the risk factors cluster into statistically meaningful groupings. PCA has previously been used in grouping risk factors by Conger, Loch and Helft 1995, Wallace, Keil and Rai 2004; and Jiang, and Klein, 1999, 2000. This initial part of the analysis reduces the candidate risk factor list to those most influential using the Kaiser (1960) criterion (explained in more detail in the next factor analysis chapter). The output of this process is to produce extracted risk factors. Once the candidate risk factors are extracted they were term simply 'risk factors' to distinguish them as confirmed from the risk factor candidate list used in the survey questions. The risk factors were then grouped as explanatory factor loadings into risk components. The aggregation of the remaining risk factors into risk components is a clustering process that is a PCA-determined method that forms cluster based on the degree of colinearity between risk factors.

The empirically derived risk components relating to questions (i) and (ii) are then interpreted in terms which are meaningful in relation to the life-cycle of the project. This reversion to the research risk construct is an important element of the research contribution, on the basis of theory and further empirical work, why the risk components observed in practice are likely to be important generally. Thus, the research seek to interpret the risk components observed in relation to project life cycle which typically involves reconfiguring the observed components to those that meaningfully relate to the risk construct, using methods that involve both appeal to theory and further statistical investigation. This method follows that of Wallace & Keil (2004) who use a socio-technical approach to help establish the (reconfigured) dimensionality of risks they observe in their own survey. Using structural equation modelling they derive a 'first order model' consisting of six dimensions to IT risks – equivalent to the research risk components – which are determined empirically. They then create a 'second order model' comprised of three risks. The relationship between first order and second order models is succinctly described in Wallace & Keil (2004) as the relationship between a measurement model and a theoretical model.

Their appeal to theory therefore establishes content validity of the risks they derive. Additionally, they conduct further statistical tests to establish criterion validity of their second order model. This research approach is identical, only this research use the term components and clusters. This research content validity is to appeal to our life-cycle risk construct to determine if a more theoretically appealing and parsimonious grouping of the empirically derived risk components can be established. This research criterion validity is to appeal to statistical correlations with mitigation strategies that will be explain next in terms of question (iii).

In question (iii), the survey participants were also asked to independently rank mitigation strategies in terms of their effectiveness as responses to IT software risk generally. The survey responses on a predetermined list of 30 strategies, chosen from the available literature that will be explain further in Chapter 8 (Risk management strategies). In linking mitigation strategies to extracted components, the statistical approach is to estimate factor scores using Bartlett's method (DiStefano et al 2009), and which are available as an option in standard SPSS packages. From this, will enable to correlate the mitigation rankings to both extracted components for risk likelihood and the impact of cost overruns to assess which risk mitigation strategies are likely to be used in practice (Rietveld & Van Hout 1993). Comments were made on the most important issues as determined by the statistical significance of the correlation coefficients we calculate. In using this approach enable to establish criterion validity for our risk construct which underpins the generality of the findings.

From the analysis of the principal component analysis and clustering of the risk factors, fuzzy modelling technique is employed to represent the risk factors. The purpose of this is to reduce the influence of subjectivity during the evaluation and assessment exercise. A model of fuzzy risk analysis is proposed to assist software practitioners and decision makers in formalising subjective thinking that are required in assessing the current risk environment of their software development in a more systematic manner.

2.9. Summary

In the end, the study are able to 'close the loop' from risk identification to risk impact to risk mitigation. This study will be able to verify if effective mitigation strategies (as determined by the survey) are correlated to groups of the risk components and, in turn, substantiate the risk construct. In this way, if the study proceed as might be expected according to the research theory and discover significant correlations that accord with the understanding of IT software risk as a project life cycle risk construct, then actions (consequences) conditioned can be observed on identification of risk likelihood and risk impact as conceptualised in terms of a project life cycle.

CHAPTER 3 PROJECT MANAGEMENT OF SOFTWARE DEVELOPMENT LIFE CYCLE

3.1. Introduction

Software development life cycle represents a set of general categories that show the major steps, over time, of a software development project. Within these general categories are individual tasks. Some of these tasks are present in most projects, while others would apply only to certain types of projects. For example, smaller projects may not require as many tasks as larger ones.

There were numerous generic software development life cycle being established in the software industries. There is no universal, standardized version of the software development life cycle. Consultant firms, as well as IS groups within organizations, develop individualised versions appropriate to their own operations and needs. They may give their versions unique names.

This chapter highlighted a few of the most common and generic software development life cycles and discussed the relevancy of the life cycle from the perspectives of project management.

3.2. Software Development Life Cycle

3.2.1. Waterfall Model

The most common and established methodology used has been called the waterfall model. The steps in this classical methodology are (Alter, 1992; 2002):-

- Definition
- Specification (requirements)
- Design
- Construction (programming and unit testing)
- Testing (system and integration)
- Installation
- Operation and maintenance

The specific steps can vary across organizations. A key characteristic of the Waterfall approach is extensive formal reviews by project team members and business management at the end of each major step. Without formal approvals, project team cannot begin on the next step of the methodology. The completion of each phase therefore represents a major milestone in the development of the system. Most Waterfall methodologies result in a lot of documentation.

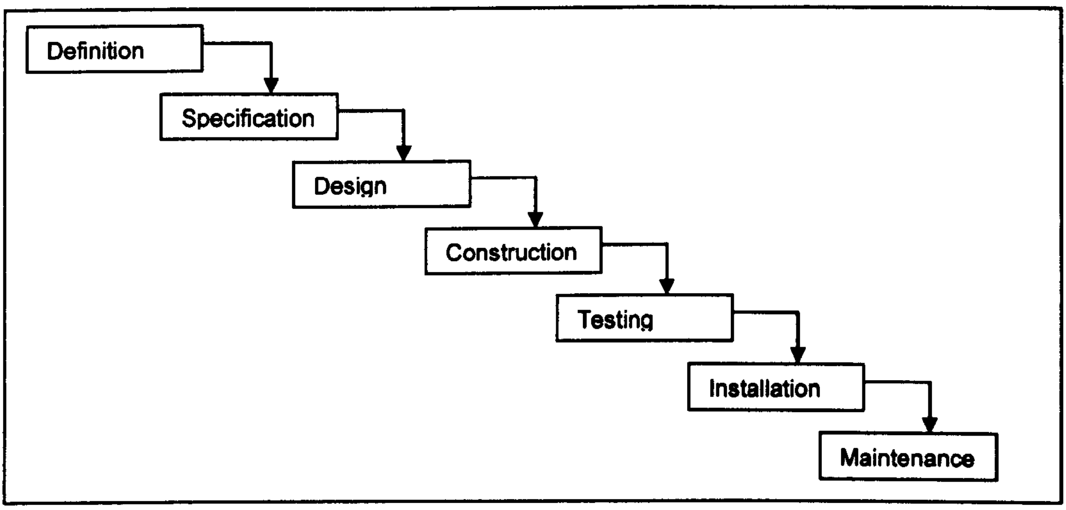


Figure 3.1 : Waterfall model approach

Although the Waterfall approach is a highly structured approach, but the highly technical terms of the stages would be only be understood by competent IS specialists and knowledgeable business managers. For the less knowledgeable business managers might have difficulty understanding the whole process of the software development. This may lead to higher risk of misunderstanding the requirements and processes, cost overrun and time overrun. Furthermore, as this traditional waterfall approach is quite lengthy and costly, so, the strong support and commitment from the management is needed. And, it is probably ideal for the business managers or management to follow the software development process in less technical terms especially the management is less knowledgeable, in order to gain the strong support.

3.2.2. Evolutionary development

The development methodology begins with only the user requirements that are very well understood and builds a first version. Often that first version is just a prototype. (Edwads et al, 1995; Capron, 1993; Alter, 2002). Analysis, design, implementation and testing are done in a free flow overlapping manner without any formal review of documents. The first version is then taken back to the customer for review and definition of further requirements. The second version is then built, and then taken back to the customer. This process continues until the customer is satisfied with a version and no further extensions are required. The risk of using this methodology is however, the visibility of the management may be limited because little intermediate documentation is produced. Lesser or little documentation could increase the failure of managing the software development project, as problems or risk associated with the project becomes difficult to trace. As a result, the application of an appropriate risk management strategy may become more complex and lengthy processes. Also, internal design is often poor for evolutionary development because the entire scope is not visible from the beginning, and continually changing a system leads to a design that is less adaptable and harder to maintain. Although, the whole idea of the review of the versions by the users, is for the benefit of the users, but, there is a risk of developing unnecessary requirements than initially anticipated within the project goals and objectives. Furthermore, as the versions go back and forth to the users until the users are satisfied, in gaining the full user satisfaction would bring the project beyond the actual deadline and budget stipulated.

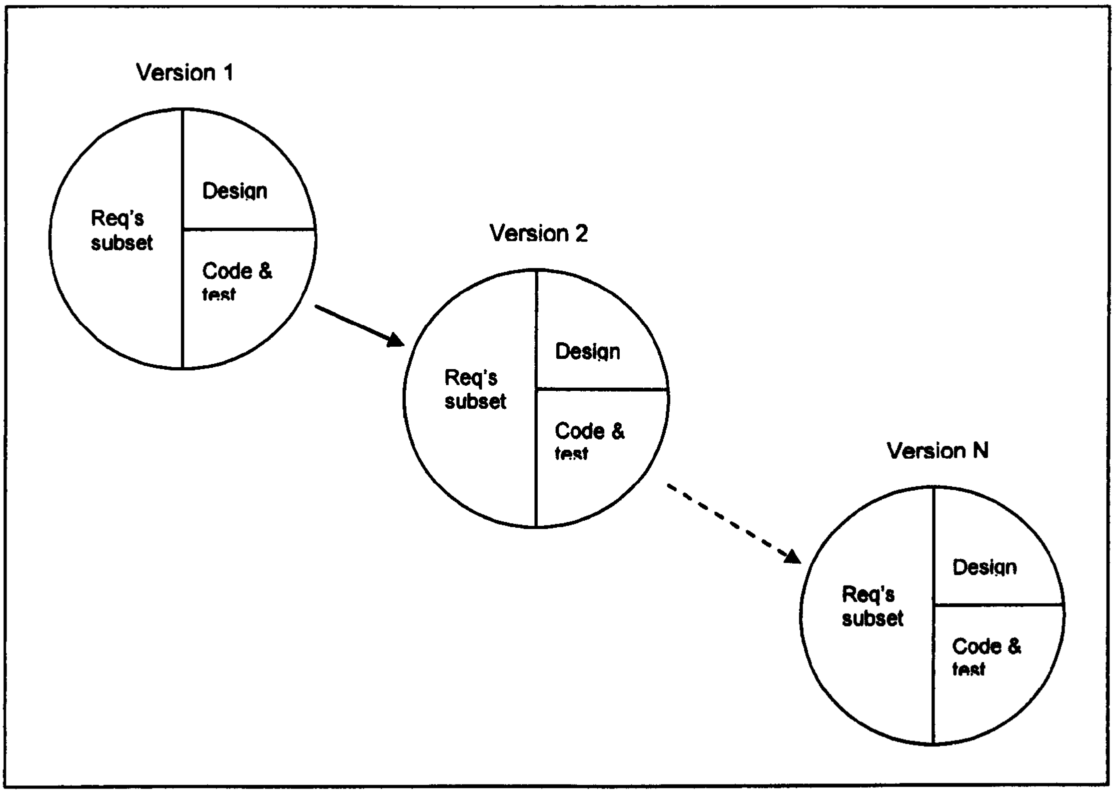


Figure 3.2 : Evolutionary development

3.2.3. Incremental development

The methodology begins with a determination of all the requirements, but only in a rough outline form. Next, those requirements are prioritised normally based upon those features that are most important from a business perspective. Because time is spent up front looking at all requirements a more appropriate overall platform, architecture and design can be selected. (Edwards et al 1995; Hussein, 1995)

After the initial requirements phase, development proceeds as in the evolutionary method. Incremental development is not as quick as evolutionary development, but attempts to avoid the design problems caused by not knowing all the major requirements initially. The risks are that the increments are based on the priorities of the requirements, and sometimes priorities may significantly change during the time of developing the increments. Also, changing of requirements or any additional requirement could mean the emergent of new risks factors. As this methodology involves the overlapping manner and iterations of the versions, and with less formal review of documentations, there is always a possibility of the same risk factors emerge a few times.

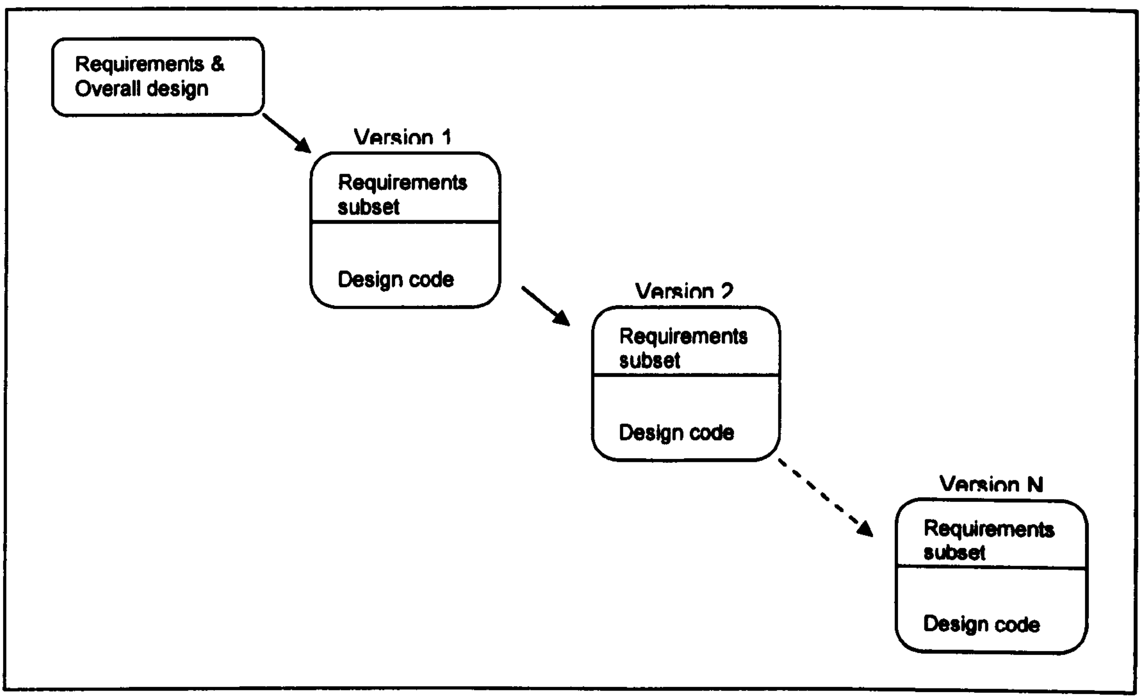


Figure 3.3 : Incremental development

3.2.4. Prototyping

It involves creating a scaled-down model of the product to be built. Instead of spending a lot of time producing very detailed specifications, the developers find out only generally what the users want.

The developers do not develop the complete system all at once. Instead they quickly create a prototype, which either contains portions of the system of most interest to the users, or is a small scale working model of the entire system. After reviewing the prototype with the users, the developers refine and extend it. This process continues through several iterations until either the users approve the design or it becomes apparent that the proposed system cannot meet their needs. If the system is viable, the developers create a full scale version that includes additional features.

In this approach, the emphasis is on producing something quickly for the users to review. After the users approve the prototype, the programmers can finish development making whatever changes are necessary for acceptable performance and for meeting organizational standards. The major risk in this method is the problem encountered developing the smaller scale model compared to the full scale version may not be the same, and risk management strategies adopted may only suitable with the lower scale version. The magnitude of the risk factors could increase significantly or not proportionate with the increase of the full scale version. Another risk is that, while most people find it very difficult to specify in great detail exactly what they need for a new software, it is quite easy for them to point out what they do not like that they can try out and use. This may lead to the prolonged of completion time for the software project and the cost originally estimated.

3.2.5. CASE (Computer-aided software engineering)

It refers to a collection of software tools used to automate some or all phases of an SDLC process. Refers to computer software systems that generally support several steps in the development process, usually including requirements, analysis, design and possibly other steps such as coding, testing, documentation and version control. (Hussein, 1995; Martin et al, 1992)

There are different types of CASE products. Some products support one particular methodology and others can support several methodologies; some support one particular computer language and some may support several languages. This CASE environment also supports a number of different types of users; project managers, designers, programmers, database administrators, testers, technical writers and so forth.

Although this product provided significant development acceleration, there is a risk that one must be careful not to let CASE replace a sound methodology, or all what will be accomplish is to build the wrong system even faster. Another significant risk is the high technical complexity of the CASE tools, as only experienced and knowledgeable project managers, developers and business managers would be able the follow and understand the development process.

The initial introduction of CASE tools into an organization also required a major commitment by management to a disciplined development methodology that is embedded within the CASE tool. If an organization changes its methodology as it introduces a CASE tool, the risk of cultural changes for an IS professional can be significant.

3.2.6. Joint Application Design (JAD)

It is a technique in which a team of users and IS specialists engage in an intense and structured processes to develop systems requirements or review a major system design deliverables. JAD session could last several hours or could be held over several consecutive days. It is often held at a location removed from the participants' usual workplace so they can concentrate on this task without interruption. (Martin et al, 1992; Martin et al, 2002)

Primary objective of JAD technique is to minimize the total time required for information gathering from multiple participants. It provides a forum for user representatives to work through areas of disagreement; this is especially important when cross-functional systems are developed. JAD session is led by a facilitator who is not only skilled in systems analysis and design techniques, but also skilled in managing group interactions. An additional benefit associated with the use of this technique, then, is the achievement of the shared understanding among business managers and IS specialists. The main risk with this methodology is the time frame taken for the process might take longer than expected. The agreements among the users, developers or IT specialist and the business managers in developing the requirements might be difficult to be agreed upon. This may also increase the chances of conflicts among the team members.

3.2.7. Rapid application development (RAD)

It is a generic term for software development methods and tools that speed up the development process. It was commonly applied to products that automatically generated code to create user interface screens both of the character type and graphical type. (Hussein, 1995; Martin et al, 1992). Products such as Clarion, Foxpro, Visual Basic and PowerBuilder are examples. Many RAD products were combined with databases such as Microsoft Access and other RAD products could interface to stand-alone database products.

Eventhough, RAD products did speed up considerably the implementation step for smaller applications, but the risk is, it did not accelerate other development steps; requirements, analysis, design and testing. For larger or more complex systems, automatic code generation is not powerful or flexible enough to meet application needs and programmers usually have to resort to traditional hand coding. The other risk is the technicality complex of the method as it is usually combined with other interfaces, which may create the emergent of new risk factors.

RAD is a methodology that works well in a business environment characterized by rapid change. The smaller design teams and shorter development times associated with RAD also can lead to lower development costs. The other risk with this methodology is that, RAD methodology is highly dependent on key users. So, if the users on the project team are not freed up to work on the project, the custom application may be produced on time, but it may not be an optimal software solution for the business.

3.2.8. Agile programming (AP)

It is a name given to a growing number of lightweight methodologies with names like Crystal (Cockburn, 2001), Scrum (Schwaber, 2001), Feature-Driven Development (Palmer, 2002). Lightweight methodologies do away with much of the SDLC process overhead that slow down developers, such as detailed formal requirements definitions and extensive documentation. The main principle for this methodology is using simple design, take small incremental steps and design as you go.

The main risk is that these development approach are based on the premise that if you hire competent programmers who always know what they are doing, then any problems they encounter are organizational and communications ones; and those are what the agile approach focuses on. Another main risk in this method is that its common feature of contact time with users where this method's key focus is on people or users, but not on processes.

3.2.9. Extreme programming (XP)

It is a software development approach initially created for small teams on risk-prone projects with unstable requirements (Beck, 1999). XP is a form of a AP based on a lightweight methodology. XP differs from most other agile approaches by being much more prescriptive. The methodology creating user scenarios and performing upfront feature testing, allows them to develop and deliver code more quickly with fewer bugs. XP is built around rapid iterations, an emphasis on code writing, and working closely with end users.

Basic practices of XP are (Hussein, 1995; Martin et al, 1992). :-

- Customers define requirements via use case scenarios.
- Simple object-oriented coding is used.
- Teams use standard names and descriptions.
- Designers write automated unit test scripts before coding.

But XP does not address downstream SDLC issues such as training and user documentation. XP requires the benefiting organization to take a very active role in the development process. Because XP requires constant communication between the benefiting and performing organization (as well as among the developers), and because communication time and traffic increases in proportion to the square of the number of communicating parties, XP is not suited to large teams (Beck, 1999).

3.2.10. Object-oriented software

An OO system begins not with the task to be performed, but with the aspects of the real world that must be modelled to perform that task. Therefore, if a firm has a good model of its customers and its interactions with them, this model can be used equally well not just the original task, but may also suited for handling other tasks. (Hussein, 1995; Martin et al, 1992). OO programming is a key part of many of the methodologies previously mentioned, such as AP and XP. OO software systems are inherently more adaptable and maintainable than traditional procedural software, and OO systems foster software reuse. Object technology promise a way to deliver cost effective, high quality and flexible systems on time to the customer. In fact, out of all the aforementioned methodology variations and development acceleration techniques, OO is the only technique proven to be almost always effective in reducing long term software development costs.

3.3. The risk of existing software development life cycle

As pointed out by Avison & Fitzgerald (1998, 2003), software development life cycle provide a systematic method of development so that progress can be effectively monitored. The provisions of checkpoints and well-defined stages in a methodology should ensure that project planning techniques could be applied effectively.

The division into phases enables managers to control and direct the activities in a disciplined, orderly and methodical way that is responsive to changes, adaptations and complications (Benediktsson et al, 2006). Phases group together directly related sequences and types of activities to facilitate visibility and control thus enabling the completion of the software project. The project life cycle thus acts as a framework focusing on the allocation of resources, the integration of activities, the support of timely decision making, the reduction of risk and the provision of control mechanisms.

However, the variation and different approaches of software development life cycle could lead to a dilemma when it comes to selecting the most suitable one for a project. At the beginning of a project the project manager is expected to commit to a development approach. This is often driven by past experience or other projects that are, or have been, undertaken by the organisation (Benediktsson et al, 2006; Davis et al, 1988; Boehm et al, 1984; Mathiassen et al, 1995). Project managers are expected to select the most suitable approach that will maximise the chances of successfully delivering a product that will address the client's needs and prove to be both useful and usable.

Often times, however, multiple methods are adopted and used together for the same project, but at different phases of the development life cycle (Khalifa & Verner, 2000; Iivari, 1996). Furthermore, there are no dominating factors have been identified as the main motivators for using any particular software development methodology, but researchers have identified number of such factors, like, the perceived effect of the development method on the quality and maintainability of the software, and a high level of control of the development process. (Khalifa & Verner, 2000; Boehm, 1988; Cerpa & Verner, 1996; Martin & Odell, 1992; Schach, 1996). Boehm & Turner (2004) define a general framework for selecting a development approach, which consist of attributes such as personnel, capabilities, requirements volatility, group culture, group size, and application criticality. Some research indicates that the use of methodology could facilitate project management, control the development process, increased productivity and quality as well as reductions in time and effort ((Riemenschneider et al 2002; Harter et al, 2000; Herbsleb et al, 1994).

However, many software development organizations do not use formal software development methodology in practice. (Vavpotic & Bajec, 2009; Avison & Fitzgerald, 2003; Fitzgerald, 1998; Riemenschneider et al, 2002). Moreover, even those that use formal software development methodology rarely follow them rigorously (Aaen, 2003; Huisman & Iivari, 2006; Fitzgerald, 1996).

The most common reasons behind the lack of usage of a formal software development life cycle is the high risk of major emphasis on technical rationality at the expense of social aspects (Fitzgerald, 1998; Wastell & Newman, 1993) and the risk of not tailoring it to the specific organization and project needs due to its rigidity and complexity. (Vavpotic & Bajec, 2009) (Avison & Fitzgerald, 2003).

The complexity on the method may lead to the risk of developing requirements to the ultimate degree, often over and above what is legitimately required. Sometimes encouraging users to create unrealistic wish lists which resulted in relatively unimportant aspects being developed to the same degree as those that are essential. Furthermore, it may fail to address the critically important social, political and organizational dimensions of the software development project and deviate from the main goals and objectives of the software project (Avison & Fitzgerald, 2003). The methodologies also require highly technical skills that can be difficult and expensive for developers and end users to learn or acquire. Moreover, the tools advocated by methodology proponents can be costly, difficult to use, yet still not deliver enough benefit. This may increase the risk of overrunning the cost and timeframe.

Other criticism of the methodologies include failing to deliver the productivity benefit as suggested by some researchers. It is said the methodology do not reduce the time taken to develop a project, rather their uses increases systems development lead times when compared with not using a methodology. This is usually because the methodology specifies many more activities and tasks that have to be taken, with the use of more diagrams or models, which could make the software project more complex. This at the same increase the risk of project overrun in relation to cost and time, which may effect the performance of the software development project.

As well as being slow, they are resource intensive, in terms of number of people required from both the development and user side; and from the point of view of the costs of adopting the methodology; the purchase costs, training, tools, organizational costs. Avison and Fitzgerald (2003) state that the main objectives of the methodology should be a systematic method can be effectively monitored, within the time frame and cost, and also well documented and easy to maintain. However, the literature suggests that the methodology is not doing what it should be, so an easily understood framework of methodology of software development project could assist the development team in the successful completion of the software development project without jeopardising the time frame and the budget stipulated. The project management framework was proposed for the software development methodology as it is the most common and widely used by managers in many projects.

3.4. Project Management of Development Life cycle

Previous research and study revealed that existing IT risk classification models have limited applicability and lacking the ability to easily communicate an organizing framework for IT risk factors. Previous study also highlighted that risk classification models should be based on clearly defined stages and concepts that are understandable, adaptable to a variety of contexts and of practical use.

PMBoK project management and project life cycles are not new concepts. Even though numerous terms and nomenclature had been used and established in relation to project management and project life cycle, there are by no means universal. Project management is essentially about managing a project from its conception to its completion through various phases and project life cycle. Life cycle terminologies like initiation phase, feasibility study, planning, development, implementation, operation and maintenance, monitoring and control, are commonly used within the project life cycle. However, there is no standard nomenclature for naming phases or life cycle, but different types of projects can and often do have more or fewer phases.

Project Management is a unique set of coordinated activities, with definite starting and finishing points, undertaken by an individual or organisations to meet specific performance objectives within defined schedule, cost and performance parameters. It is a work effort made over a period with a start and a finish to create a unique product, services and results within clearly specified time, cost and quality constraints. The project is completed when the goals and objectives of the project are accomplished. Sometimes projects can be ended when it is determined that the goals and objectives cannot be accomplished and the project is cancelled or terminated. The projects usually define as a steady progression of a project from its beginning to its completion. The pattern of the life cycle from slow-rapid-slow progress towards the completion of a project is quite common in project management. For most part, it is a result of the changing levels of resources and time spend during the successive stages of the project life cycle. This changing pattern requires a steady and coordinated progression of stages throughout the project life cycle, which is a key concept of project management life cycle.

The basis of using the PMBoK project life cycle is that categories, groupings and dimensions of risk factors based on project management perspectives could potentially provide a broader framing – and hence be more widely applicable - for thinking about what risks might be targeted for risk mitigation. A further motivation for employing a project management perspective is that resulting risk categorisations will be placed in context of the stages, processes and activities within the software development project. As such, risk management becomes engaged with project management in a clearer manner.

Kuruppuarachchi et al (2002) stress that, although many factors are involved to achieve IT project success, having a detailed action with identifiable stages in a project lifecycle context could increase the chances of success. Meredith & Mantel (2007) also highlighted that organising risk factors in a project management framework facilitates a focus on roles and responsibilities and allows for the coordination and integration of activities for regular monitoring and aligning with the project goals. Such taxonomy would then, arguably, better enable management to identify and manage risk as they emerge with project stages and more closely reflect project activity, allow more accurate assessment of the level of impact, and facilitate the identification of the most appropriate risk mitigation strategies.

Table 3.1 and Table 3.2 gave some summary of what is involved in the project management life cycle, the stages, the activities and processes associated with it and the expected end product.

Table 3.1 : 4 phases of Project Management
(Project Management (3rd edition)- Harvey Maylor – 2003; Pearson Education Limited.)

PHASE	KEY ISSUE	FUNDAMENTAL QUESTIONS
Define the project	Project and organization strategy. Goal definition	What is to be done? Why is it to be done?
Design the project process	Modelling and planning Estimates and resource analysis Conflict resolution and justification	How will it be done? Who will be involved in each plan? When can it start and finish?
Deliver the project	Organisation and control Leadership and decision making Problem solving	How should the project be managed on a day to day basis?
Develop the process	Assessment of process and outcomes of the project Evaluation Changes for the future	How can the process be continually improved?

Table 3.1 and Table 3.2 shows the general phases and stages of project life cycle. It also shows the relevant goals and issues, or input and the output expected for any particular cycles or stages. The issues of what the project is all about, how it should be done, who should do it or what is the expected outcome from it are crucial issues.

Table 3.2 : Stages of project life
E-books : PMP in depth :
Project Management Professional study guide for PMP and CAPM exams.
Boston, USA : Course Technology Incorporated 2006.

STAGE	MAIN GOAL	MAIN OUTPUT
Initiating	Authorize the project	Project charter and preliminary project scope statement
Planning	Plan and schedule the work to perform the project	Project management plan that contains subsidiary plans, such as scope management plan and scheduled management plan
Executing	Perform the project work	Project deliverables; product; service; results.
Monitoring and controlling	Monitor the progress of the project to identify the variance from the plan and to correct it	Change requests and recommendations for preventive and corrective actions
Closing	Close the project formally	Product acceptance and closure

The definitions stress the achievements of predetermined project objectives, which normally refer to scope, quality, time, cost and participant satisfaction, and directly links them to the project life cycle. The PMI (PMI 2000) defines the project life cycle as the steady progression of a project from its beginning to its completion. The life cycle of a project is divided into phases and stages. However, some phases of most projects involve iterations to a greater or lesser degree depending on the type of project. The cycles within cycles are common to many other project processes. Indeed for large projects, the project life cycle can be replicated within each phase, as each of them becomes a mini project in its own right.

3.5. Relationship of different development models and overall life cycle.

Chapter 3 highlighted some of common and generic software development life cycle used in the software industries. The variation and the different life cycle approach could make the selection of the most suitable lifecycle for a software project very difficult. There are no dominating factors that contribute to the selection of any particular method. Sometimes, multiple methods had been used together for the same project, driven by past experience or previous projects. However, literature suggests that many organizations did not use the formal software development methodology rigorously in practice. This lack of usage usually due to the risk of emphasizing on the technical design requirements and not focusing to the specific project objectives. These methodologies also require highly technical skills that are difficult, expensive and time consuming for developers to learn and acquire.

Most of the software development life cycle mentioned did not really specify or stressed the importance of defining the project scope or objectives, project planning or project control during its life cycle. The existing software development lifecycle tends to focus mainly on the technical and design requirements of the software project, and involve iterations of processes based on working model or prototype versions. However, project management concept usually follows a life cycle thorough a coordinated activities from the beginning to the end. This includes determining the project objectives from the beginning, the feasibility study, the project planning, the monitoring and operation. The project scheduling, the project cost and performance parameters were also identified from the early stage and closely monitored throughout the project.

As the existing software development life cycle focused mainly on the technicality of the requirements, the development may overlooked other important part of a project life cycle like the early phase before the design requirements or the later stages like the implementation phase. Any failures, setbacks in the software may only be traced back only to the design phases, whereas, the real problems might be in the earlier stages like the project planning. As the existing software development did not also stressed the important of the later stages like operation or maintenance, the system may be difficult for the team to maintain or operate. Furthermore, the iteration processes on most of the software development makes it more difficult and complex to control.

Ideally, the development life cycle should be a systematic method of development that can be effectively monitored within the time frame and cost, well documented and easy to maintain. The software development life cycle should provide as a framework focusing on the provisions of checkpoints and well-defined stages in ensuring that project planning techniques could be applied effectively. The division into stages or phases directs the activities and processes in a disciplined and orderly manner which could act as a control mechanisms for the managers in making timely and effective decision making.

3.6. Summary

From authorization to completion, a project goes through a whole lifecycle that includes defining the project objectives, planning the work to achieve those objectives, performing the work, monitoring the progress, and closing the project after receiving the product acceptance. In most of software development lifecycle mentioned, most of development life cycle did not have a very specific start and end date, but have a few iteration processes and very brief requirements. Based on the definition of project, project management, processes and project lifecycle, it is clear that most of the generic software development life cycles mentioned do not really consistent within the perspectives of project management definitions. Hence, the research produces a new framework of the software development life cycle. This new framework based on project management perspectives is explained further in the next chapter.

CHAPTER 4 RISK FACTORS LITERATURES & FRAMEWORK

4.1. Introduction

Literatures of risk factors of software project shows that a number of risks list had been generated by many researchers in this area. Most of the risks lists being conceptualized into groupings and dimensions on the basis of the risk factors characteristics and common themes. This research proposed to categorize the risk factors into a new framework based on the project management life cycle.

In this chapter, the focus is on extracting the generic risk factors from the literature reviews of software project risk which are common to most projects. The risk factors extracted were organized into the relevant stages of the project management perspective to make them more useful and meaningful for the managers and practitioners.

4.2. Risk definition

One way of defining risk is that the risk is a problem or threat that has not happened yet. While this may be a bit simplistic, it does get to the core of the issue a company of a project manager faces is that, what are the problems might be encountered while performing this project and how do to manage them. Chapman and Cooper (1983), defines risk as exposure to the possibility of economic or financial loss or gains, physical damage or injury or delay as a consequence of the uncertainty associated with pursuing a course of action. In general, unexpected events occur in projects and may result in either positive or negative outcome that deviates from the project plan (Ahmed et al; 2007).

A more technical interpretation of risks in projects can be defined as the chance of an event occurring that is likely to have an impact on project objectives and is measured in terms of likelihood and consequence (Carter et al 1993; Chapman 1998; Baccarini et al 2004). Risk factors can also be interpretated as a condition that can form a serious threat to the successful completion of an IT project (Schmidt et al,2001; Conrow & Shishido, 1997; Huang & Han, 2008; Wallace et al, 2004). Whereas software risk management can be defined as an attempt to formalise risk oriented correlates of success into a readily applicable set of principles and practices (Ropponen and Lyytinen, 2000). Dey et al (2007) refers risks as future conditions or circumstances that exist outside of the control of the project team that will have an adverse impact on the project if they occur.

Even though there are several definitions of risk available in literatures and previous research study, most definition usually referred risks as an exposure to losses in a project and as a probability of losses in a project.

As most literatures referred risk as an exposure to losses in a project and as a probability of losses in a project, this research used the definition of risk, as events that occur in a software project and resulted in a negative impact on the project.

4.3. Risk Management Process

Project risk management has been seen as a process to manage events which have an effect on project's objectives, such as cost, time, scope or quality objectives. (Cooper et al, 2005; Olson, 2007; Perminova et al, 2008). The management of risks must take into consideration the evolving and dynamic nature of the projects and, the different degrees of uncertainty through time (Chapman & Ward, 2003). Jaafari (2001) and Ward and Chapman (2003) expressed the importance of considering risks and opportunities during the risk analysis proses. Ward and Chapman proposed an approach called uncertainty management which considers the positive and negative consequences of uncertainty (Chapman & Ward, 2003; Chapman, 2003). They argue that the word 'risk' already has a negative meaning, and may complicates the exploration of opportunity during the risk identification and analysis process. However, project risk management has a strong orientation towards the negative effects. (Hillson, 2002; Jaafari, 2001; Pellgrinelli et al, 2007; Ward & Chapman, 2003; Zhang, 2007). Kristensen et al (2006) propose that risk control strategies be grouped into three categories:-

a. Risk based approach, which focuses directly on risks found and developed from the analysis of these risks and their possible solutions. The strategies include avoidance, reduction, transfer and retention.
b. Precautionary approach, is based on continuous project monitoring, continuous search of risk and the development of substitutes. It is not based on formal risk assessment.
c. Discursive approach, is oriented towards people with the intention of building confidence through reduction of uncertainties, involvement of affected people and accountability.

This classification takes into account treatments looking to control the negative consequences. Chapman and Ward consider this kind of plans to be a reactive response to the uncertainty. It is also necessary to establish plans with proactive responses in order to treat opportunities. (Chapman & Ward, 2003).

Jaafari (2001) highlighted that conventional project risk management approaches did not explicitly consider strategic and holistic risks. The approaches were orientated towards the identification and analysis of risks such as technical, operational, cost and schedule. But, the PRAM Guide (APM, 2004) and Managing successful Project with PRINCE2, consider strategic issues and the wider perspective of the organization (OGC, 2002). However, the project risk management limitation is its low implementation in the industry (Kwak & Stoddard, 2004; Uher & Toakley, 1999). Kwak and Ibbs (2004) also showed that risk management is the least practiced discipline among the knowledge areas in project management. In addition, Hobbs and Aubry's (2006) results show that only 29% of the project offices studied consider managing a risk database to be an important function. Elkington & Smallman (2004) found that there is a strong link between the amount of risk management undertaken and the level of project success; more successful projects use more risk management. However, the finding can also be accredited to more thorough project management done due to the risk management process.

Number of risk management tools, techniques and management approach have been introduced and established, though these tools and techniques have benefits and limitations. There is no actual dominating risk management strategy. Applications guides have also been produced by the professional institutions and the standards bodies to devise a general process for managing risks in projects. This is shown in Table 4.1 below:-

	Project Risk Analysis & Management (PRAM Guide) (APM, 2004)	**PMBoK guide**	**Management of Risk Guideline: Guidance for practitioners (OGC, 2007)**	**The Orange book; management of risk – principles and concepts (HM Treasury, 2004)**
1	Iniatiate	Risk management planning	Identify	Establishing context
2	Identify	Risk identification	Assess	Identifying risks
3	Assess	Qualitative risk analysis	Plan	Accessing risks
4	Plan responses	Quantitative risk analysis	Implement	Addressing risks
5	Implement responses	Risk response planning	Communicate	Reviewing and reporting risks
6	Manage process	Risk monitoring and control		Communication and learning

Table 4.1. Comparison of project risk management processes.

The risk management processes shown can provide a generic structure which can be tailored considering the needs of each organization and the characteristics of projects. But, there may still be deviations in the practice which must be solved without following the standard risk management procedures (Hallgren, 2007, Payne & Turner, 1999).

The PRAM guide is specifically developed to be used in the project management domain (APM, 2004). It is oriented to avoid or decrease threats and to exploit or make the most of opportunities (Chapman and Ward, 2003). It focuses on the project-specific issues and also considers how the risk management process at project level connects to strategic and corporate goals. The PMBoK guide is written specifically to be applied in the project risk management field and has a linear framework composed of inputs, processes and outputs (PMI, Cooper et al, 2005). The PRAM and PMBOK Guide introduce the steps to following a risk management process from the context definition step to the risk control and monitoring step. They provide a list with descriptions of tools and techniques that focus directly on project issues.

The Management of risk guideline by OGC is develop for public and private sector organizations and deals with risks affecting the organization's success in a positive and negative manner (OGC, 2007; Cooper et al, 2005). It highlights the importance of identifying the interdependencies linking the project to its context and the organization maturity model depending on the level of risk management implementation. The Orange book – management of risk, by the HM Treasury consider risks in a corporate context for managing strategic and organizational risks. In spite of not being developed for a project management context, the guideline implement risk management processes at all levels of the organization. The structures highlight the role of different organizational actors during the management of risks.

Although the process models of each guidelines may differ in detail and terminologies, they all tend to show a series of discrete activities, and tend to agree on the key activities. The similarity and consistency across the processes would indicate the consensus regarding the way risk management ought to be conducted. This include :-

- Identification of the risk issues
- Analysis and assessment of the risks for their potential impact on the project.
- Deciding whether anything can or should be done about the identified risks.
- Developing responses, where required, to the risk issues; some may be proactive while others may be in the form of a contingency.
- Monitoring the situation
- Reassessing the situation in the light of actions taken or risks materializing.

It is generally accepted that there are a few course of action that can be adopted in response to perceived risks as shown by the risk management processes or guidelines. Once the principal sources of the risks being identified, they need to be analysed and assessed for their effect on the project. This may involve both analytical thinking and making subjective judgements about the future. Having assessed the perceived risks and decide on how to handle them, it is necessary to continuously monitor all changes in circumstances that could affect the risk either by making it more or less likely to materialize or altering its effect.

4.4. Extraction of IT risk factors

Many studies have proven that a proper management of software risks affects the success or failure of a project. (Wen & Sun,2007; Jiang & Klein,2000; Wallace & Keil, 2004). Identifying the software risks that negatively affect the project performance, should be well controlled in order to improve the project performance. Failure to understand, identify and manage these risks is often cited as a contributing factor in IT project failures.

The extent of IT literature has produced a number of conceptual frameworks to explain different types of software development risk, risk management strategies and measures of software project performance (Kwan et al,2007; Nidumolu,1996; Wallace & Keil,2004). Many studies suggest that failure to manage risks causes common problems such as cost overruns, unsuitability for intended task, unmet user requirements and schedule overruns (Keil et al,2002). Clearly, systems development projects can present serious risks to the well being of an organisation.

The presence of various risks factors in software projects and the need to manage these risks is well documented in the IT literatures. Through the literature reviews of previous research of software project risk, extraction of software risk influencing factors was done. Since there is a significant number of overlapping of risk factors within the other researchers; the most commonly cited risk factors were extracted. The whole purpose of this extraction is to list out the risk factors for the questionnaire design and developing a classification framework. Apart from the literature reviews, the list resulted from the extraction was also supplemented with other risk factors which the author believed was significant based on the author's knowledge and informal discussions with colleagues. The list of risk factors and their related literatures was shown in Table 4.2 :-

Table 4.2. : Research undertaken in IT risk management

YEAR	RESEARCHER	RESEARCH AREA	Risk list	Point of view
2008	Mark Keil et al	The influence of risk checklists on software practitioner risk perception and decision-making, and, the influence of role (inside project manager vs. outside consultant) on software practitioner risk perception and decision-making. Their research use risk checklists define into 13 categories of risk factors; Corporate environment Ownership Relationship	Lack of users involvement Inadequate validation of requirement Inadequate resource estimate Users resistant to change Conflict between users Lack of top management support for the project Undefined project success criteria Conflicting system requirements Unclear project scope/objectives System requirements not adequately identified Unclear system requirements Incorrect system requirements Gold platting or unnecessary requirement Ill-defined project goals Users lack understanding of system capabilities and limitations Difficulty in defining the inputs and outputs of the system Inadequately trained development team members Lack of commitment to the project among development team members	Project Manager and outside consultants

		management Requirements Funding Scheduling Development process Personnel Staffing Technologies External dependencies Planning Project management	Inexperienced team members Frequent conflicts among development team members Frequent turnover within the project team Development team unfamiliar with selected development tools Team members not familiar with the task(s) being automated Team members lack specialized skills required by the project Project involves the use of new technology High level of technical complexity Highly complex task being automated Project affects a large number of user departments or units One of the largest projects attempted by the organization Large number of links to other systems required Immature technology Project involves use of technology that has no been used in prior projects Lack of an effective project management methodology Inadequate estimation of project schedule Lack of people skills in project leadership Project progress not monitored closely enough Poor project planning Project milestones not clearly defined Inadequate estimation of project budget Ineffective project manager Inexperienced project manager Ineffective communication Resources shifted from the project due to changes in organizational priorities Change in organizational management during the project Corporate politics with negative effect on project Unstable organizational environment Organization undergoing restructuring during the project Dependency on outside suppliers Many external suppliers involved in the development project	
2007	Weng Ming Han et al	The MANOVA analysis of probability of occurrence and impact of software risks on project performance within six dimension; User Requirement Project Complexity Planning and control Team Organization environment	User resistance to change Conflicts between users Lack of cooperation from users Systems requirements not properly identified Unclear system requirements Incorrect system requirements Project involved used of new technology High level of technical complexity Immature technology Lack of effective project management methodology Project progress not monitored Inadequate estimate of resources Poor project planning Project milestones not clearly defined Inexperienced project manager Ineffective communication Inexperienced team members Inadequately trained team members Team members lack skills required for project Changes in organizational mgt during project Unstable organizational environment Organization restructuring during project	Analysis of dataset from software projects
2007	Helio Costa et al	Technique for evaluating risk levels in software projects through analogies with economic concepts, which allows a manager to estimate the probability distribution of earnings and losses incurred by an organization in relation to its software project portfolio. The research categorize the risk factors into two categories;	Unclear requirements Inadequate validation of requirements Misunderstanding of requirements Incorrect requirements High complexity Large project size Performance failure Large number of interfaces Programming language not suitable Testing plan Time for testing Insufficient unit testing Integration into existing system Detail work breakdown structure Project milestones not established Project management tools Contingency plan	Project managers

		Systemic risk Specific risks Systemic risks look into how external issues may affect an organization. Specific risks for internal factors that can affect its performance.	Inexperienced project managers Mechanism for quality procedure Configuration manager and system configuration Inappropriate development methodology Inadequate documentation Tracking of progress Team trained skills Ineffective communication	
2007	Tesch et al	IT project risk perspective of project management professionals (PMP) *The study did not categorize the risk factors into any grouping.*	Lack of top management commitment Failure to gain user commitment Misunderstanding of requirements Lack of adequate user involvement Lack of required knowledge/skills required for project Unclear scope and objectives New technology Failure to manage end user expectation Insufficient/inappropriate staffing Conflict between user departments Poor communications Lack of leadership Poor project management Excessive schedule pressure High complexity project Inadequate documentation Gold platting and unnecessary requirements	Views of project managers
2006	Eun Hee Kim et al	Explores relationship among major system development risks, and the stages in which individual risks have critical impact on development project, using association rule mining. *The study did not categorize the risk factors into any grouping.*	Requirement definitions are not clear. Unnecessary requirements are present. Project uses immature or state-of-the-art techniques for hardware, middleware, languages, methods, *etc.* Complexity of function model is very high Complexity of data model is very high. Project develops wrong function. Defaults in system performance Large number of interface. System failure and breakdown Defaults in interface with external systems are present Unanticipated difficulties in user interface development appear. Capability of staff cannot meet required level Training for staff is inadequate to meet required level Communication channels among staff do not operate properly. Frequent changes in project staffing Inexperienced project manager Development productivity is poor owing to low commitment of staff. Commitment of customer is insufficient. Conflicts on customer side over project issues are present. Disagreement within development team. Project size/complexity is underestimated. Project budget and schedule are unrealistic. WBS/work plan is inadequate. Project does not progress as planned. Selection of development method or tool is inappropriate. Major processes for system development project (*e.g.*, quality/quality assurance management, configuration/change management, requirement management, risk management, *etc.*) are not defined. Inadequate documentation	
2006	Perera et al	Prompt lists tool for risk management in Sri Lankan software industry The research categorize the risk factors into 14 categories;	Misinterpretation of system requirements Unclear requirement Unclear scope objectives User resistance to change Project complexity Project size Internal interfaces Time for testing	Software companies in Sri Lanka

		Requirement analysis Designing Coding and unit testing Integration and testing Maintenance Work environment Development process Development system Resources Legal risk Program interfaces Management Financial Political	Unsuitable language Inadequate intergration and configuration Insufficient monitoring mechanism Regular update against goals Inadequate documentation Security Programme language obsolete Hardware software resources Lack of experience Training Inexperienced project managers Lack of contingency plan Lack of quality procedure Ineffective communication Inadequate estimate Staff turnover	
2005	Wat &Ngai	Risks analysis to e-commerce development. The research categorize the risk factors into 10 categories; Resource risk Requirement risk Vendor quality risk Client server security risk Legal risk Managerial risk Outsourcing risk Physical security risk Cultural risk Reengineering risk	Hackers Unauthorised acces Threat to sabotage Inadequate back up systems Project complexity Technology newness Natural disasters caused equipment failiure Wrong function Wrong user interface Unrealistic schedule Unrealistic estimate Personnel shortfalls Lack of experience expertise Lack of top management support Poor project planning Unclear project objectives Lack of contingency plan Organizational restructuring Loss of data control	Survey of EC practitioners in Hong Kong
2004	Susan Sherer & Steven Alter	Reviews of previous research models of IT project risks And proposed a work system framework to organized the risk factors, but without any empirical research data The framework was organized into 9 work system elements :- Work practices Participants Information Technologies Products & services Customers Environment Infrastructure Strategies.	The study uses the collection and risk lists of previous researchers in the likes of Boehm (1991), Barki et al (1993), Keil et al (1998), Jiang & Klien (1999; 2000); Ropponen & Lyytinen (2000); Schmidt et al (2000), Keil et al (2002), Addison (2003), Wallace et al (2004) and Wallace & Keil (2004)	
2004	Kwok Tai Hui & Biau Liu	Bayesian belief network model to evaluate risk and impact in software development projects. *The study did not categorize the risk factors into any grouping.*	Staff experience shortage Schedule pressure Lack of staff commitment Low productivity Inaccurate cost estimate Large and complex interface Incapable project management Lack of senior management support Immature technology Inadequate configuration control Lack of experience of project manager Lack of experience of project environment Excessive schedule pressure Large and complex project	Survey of project managers

2004	Wallace & Keil	Identification of risks that posed threat to successful project outcomes. The research categorize the risk factors into 4 categories; Customer mandate Scope and requirement Environment Execution	Lack of user participation Users resistant to change Conflict between users Users not committed to the project Lack of cooperation from users Lack of top management support for the project Lack or loss of organizational commitment to the project Undefined project success criteria Conflicting system requirements Unclear project scope/objectives System requirements not adequately identified Unclear system requirements Incorrect system requirements Ill-defined project goals Users lack understanding of system capabilities and limitations Difficulty in defining the inputs and outputs of the system Inadequately trained development team members Lack of commitment to the project among development team members Inexperienced team members Frequent conflicts among development team members Frequent turnover within the project team Development team unfamiliar with selected development tools Team members not familiar with the task(s) being automated Negative attitudes by development team Team members lack specialized skills required by the project Project involves the use of new technology High level of technical complexity Highly complex task being automated Project affects a large number of user departments or units One of the largest projects attempted by the organization Large number of links to other systems required Immature technology Project involves use of technology that has no been used in prior projects Lack of an effective project management methodology Inadequate estimation of project schedule Lack of people skills in project leadership Project progress not monitored closely enough Inadequate estimation of required resources Poor project planning Project milestones not clearly defined Inadequate estimation of project budget Ineffective project manager Inexperienced project manager Ineffective communication Resources shifted from the project due to changes in organizational priorities Change in organizational management during the project Organization undergoing restructuring during the project Dependency on outside suppliers Many external suppliers involved in the development project	Project managers
2004	Wallace et al	Investigation of dimensions of risk and an exploratory model, on the software project performance. The research categorize the risk factors into 6 categories; Organization environment risk User risks Requirement risks Project complexity risks Planning and control risk Team risk	Change in organizational management Organization undergoing restructuring during project User resistant to change Conflict between users User not committed Lack of cooperation from users Continually changing system requirements System requirement not adequately identified Unclear system requirements Incorrect system requirements Project involved use of new technology High level of technical complexity Immature technology Lack of effective project management methodology Project progress not monitored closely Inadequate estimate of resources Poor project planning Project milestones not clearly defined Inexperienced project manager Ineffective communications	Project managers

			Inadequately trained development team members Inexperienced team members Team members lack specialized skills required for project	
2003	T.Addison	A study to determine the opinion of expert practitioners of the most important risks in the development of E-commerce projects, where the respondents were mainly project managers from South African software houses. Various academics and users of e-commerce systems also contributed to the survey. The Delphi technique was used to gather the data and to rank the risks. *The study did not categorize the risk factors into any grouping.*	Misunderstanding system requirements Absence of declared business benefit Too narrow focus on IT project issues Inadequate security features Lack of top management commitment Failure to manage end user expectation Insufficient procedures to ensure security, integrity Lack of user commitment and involvement Inadequate testing Complexity of interfaces Absence of regular reviews against goals High and unplanned support and maintenance costs Dependence on multiple products and suppliers Applying incorrect technology Inadequate methodologies Loss of data during conversion	Project managers and users
2003	Keil et al	Logistic regression to model relationship between various project management constructs, project escalation and risks. The research categorize the risk factors into 4 categories; Project planning Project specification Project estimation Project monitoring	Project milestone not identified Project activities not planned Project size Project time and scheduling Project complexity Unclear scope Project progress not monitored No regular updating against goals No project control mechanism Senior management did not monitor project	Survey of IS audits and control professional
2002	Keil et al	Reconciling user and project manager perception of IT project risk using Delphi study *The study did not categorize the risk factors into any grouping.*	Lack of top management commitment Misunderstanding of requirements Not managing change properly Failure to gain user commitment Lack of effective project management skills Lack of adequate user involvement Failure to manage end user expectations Lack of effective project management methodology Unclear scope Lack of knowledge skills in project personnel Introduction of new technology Inappropriate staffing turnover Conflicts between users Number of organizational units involved Lack of effective development methodology Inproper definition of roles and responsibilities Lack of available team personnel Poor team relationships Inadequate planning	Users and project managers
2001	Schmidt et al	List of common risk factors in software project using ranking type Delphi Survey Their research use risk checklists define into 13 categories of risk	Change in organizational and management structure Lack of top management commitment Conflicts between users Failure to manage user expectations Not managing change properly Lack of effective project management skills Lack of project management methodology Inproper definition of role and responsibilities	Project managers

		factors; Corporate environment Ownership Relationship management Requirements Funding Scheduling Development process Personnel Staffing Technologies External dependencies Planning Project management	Wrong development menthodology Unclear scope objectives Number of units involved Misunderstanding of requirements New technology Bad estimation Ineffective communication Dependence on outside consultant Lack of user involvement Unclear system requirement Inadequate requirement validation Inadequate resource estimate Inadequate skills in development Changes of personnel or staff turnover Inadequate understanding of technology Inadequate skills of project planning	
2001	Jiang & Klien	Explore the types of risks factors in IT project encountered, impact on different success categories, and types of strategies deployed to mitigate known risks. Identified 6 project risk categories:- Technological acquisition Application size Teams' application expertise Users' support Role definition Users experience	Technological newness Large project size Lack of team general expertise Lack of team expertise with task Lack of team development expertise Lack of user commitment Insufficient resources Unrealistic budget Unrealistic scheduling Lack of clarity of role definitions Ineffective communications between project stakeholders Large number of link to existing systems Overall knowledge of organization operations Expertise in specialised skills for the project Users negative attitude and opinions Users not ready to accept the new system Users' not familiar with development tasks	Survey of project managers
2000	Cule et al	Strategies for heading off IS project failures Categorise the risk factors into 2 main categories of Inside risk and Outside risk. For the Inside risk, there are 2 subgroup of Task risk and Self risk. For the Outside risk there are 2 subgroup of Client risk and Environment risk	Not managing change properly Lack of effective project management skills Lack of effective project management methodology Inproper definition of roles and responsibilities Misunderstanding the requirements Poor control Poor risk management Wrong development methodology Bad estimation New technology Lack of skills required Poor team relationships Insufficient staffing Dependent on outside consultants Lack of management commitment Failure to gain user commitment Conflicts between user department Failure to manage end user expectation Lack of cooperation from users Unclear scope objectives Number of units involved Unrealistic schedule Changes in organization management Lack of control and coordination Unstable corporate environment Changes of organization priorities	IS project managers
2000	Ropponen & Lyytinen	Addressing the components of software development risks using principal component analysis and one-way ANOVA. Their research use risk checklists define into 6	Personnel shortfalls Unrealistics schedules Unrealistic budgets Developing wrong software function Developing wrong user interface Requirement changes Poor system performance Project complexity	Project managers

		categories of risk factors; Scheduling and timing risks Functionality risks Subcontracting risks Requirements management Resource usage Performance risks	New technology Unsuitable development methodology Lack of skills Inexperience project manager	
1999	James Jiang & Gary Klien	Exploring the relationship between risk and different aspect of project success, where four IS success measures were found to relate to different risk factors; Development process System use satisfaction System quality Organizational impact.	Technological newness Large project size Lack of team general expertise Lack of team expertise with task Lack of team development expertise Lack of user commitment Insufficient resources Unrealistic budget Unrealistic scheduling Lack of clarity of role definitions Ineffective communications between project stakeholders Large number of link to existing systems	Survey of IS project managers
1998	Keil et al	Framework for identifying software project risk with different ranking results among project managers. Their research use risk checklists define into 4 categories of risk factors; Requirement Customer mandate Environment Execution	Lack of top management commitment Failure to gain user commitment Misunderstanding of system requirements Lack of adequate user involvement Failure to manage end user expectations Lack of required knowledge/skills Introduction of new technology Inappropriate staffing Conflict between users	Project managers of Finland, Hong Kong, US.
1994	Leslie Willcocks & Catherine Griffiths	Review of existing research and framework and put forward complementary risk profile in large scale IT projects *The study did not categorize the risk factors into any grouping.*	Large number of divisions Unrealistic project scheduling Large project size Too focus on IT Overlooked management issues Insufficient IT expertise Competitors actions High complexity Newness and changing technologies Unclear objectives Market demand Management support User commitment Number of units involved Project team experience Staff turnover/stability Technical performance User/market acceptance	Review of previous research
1993	Barki et al	Assessment of software risks resulted in five category of risks. Their research use risk checklists define into 5 categories of risk factors; Technological newness Application size Expertise	New technology Dependence on external vendors Number of users Number of people on the team Project size Team's lack development expertise Team's lack expertise with task and application Number of links to existing systems Technical complexity Organisational changes Conflicts Lack of clarity of role definitions	Survey on software projects

		Application complexity Organizational environment	Insufficient resources Task complexity	
1991	Boehm	Surveyed experienced project managers and produced risks lists *The study did not categorize the risk factors into any grouping.*	Personnel shortfalls Unrealistics budgets Unrealisitic schedules Developing wrong functions Developing wrong user interface Gold platting Unclear requirement System performance shortfalls Shortfalls in externally performed task Straining computer science capabilities	Project managers

4.5. Classification of risk factors

Many of the studies and research to date examine the risks from a broad aspect of software and IT development project. Previous researches discuss the issues of risks in more common and generic rankings or groupings of risks. Most studies explore the studies of risks using particular modelling techniques, and discuss the relationship of the software risks with the overall project performance and project success.

In a literature study by Susan and Alter (2004a & 2004b), as elaborated previously in the methodology chapter, her study highlighted the lack of easily communicate organizing framework for IT risk factors. Her reviews were based on previous researchers work without any empirical study, surveys, interviews or research data analysis to support and validate her arguments. But, her reviews suggested that organizing risk factors into a general but adaptable model could make the IT risk factors more accessible and usable by managers. Her reviews suggested that, better ways of describing risk and relating it to everyday business projects and operations may enable business and IT professionals enhancing communication and better collaboration in attempting to reduce IT related risk factors.

Other researchers have also organised and categorised the risk according to dimensions, task, structure, element and attribute. As shown in Table 4.1, Weng Ming Han et al (2007) had organised the risk factors into six dimension of user, requirement, project complexity, planning and control, team, and, organization environment, They used this six dimension for MANOVA analysis of probability of occurrence and impact of software risks on project performance. Mark Keil et al (2008) had also categorises risk factors in software project for their research, which they categorised them into 14 categories of risk factors. Helia Costa et al (2007) in their study of the techniques of evaluation of risk level of software project through analogies of economic concepts had also categorised their risk factors into two categories of specific risks and systemic risks. Other researches that have categorised their risk factors can be referred from Table 4.1, which include Perera et al (2006) with 14 categories, Wat and Ngai (2005) with 10 categories, Wallace et al (2004) with 6 categories, Keil et al (2003) with 4 categories, Schmidt et al (2001) with 13 categories and Ropponen and Lyytinen (2000) with 6 categories. There are some common categories among the researchers but none of the risk factors was categorised based on project management perspectives.

4.6. The framework of software project risk

The dimensions and categories of risk factors in software project previously studied did not really focus on the cycles, processes and activities within the software development project, but more on the characteristics and common groupings of the risk factors. The lack of focus on the cycles, processes and activities of the software project could make the management of the software project, the risk factors involved and the related risk management strategies difficult and complex. It is going to be a rather difficult task for the managers to pin-point risk factors to its most relevant stages, assessing the level of impact, or even identifying the most appropriate risk management strategies. Using the dimensions or categories identified from previous researches could make the risk management process time consuming and complex as the process could go back and forth because of the connection of the risk factors with more than one stage.

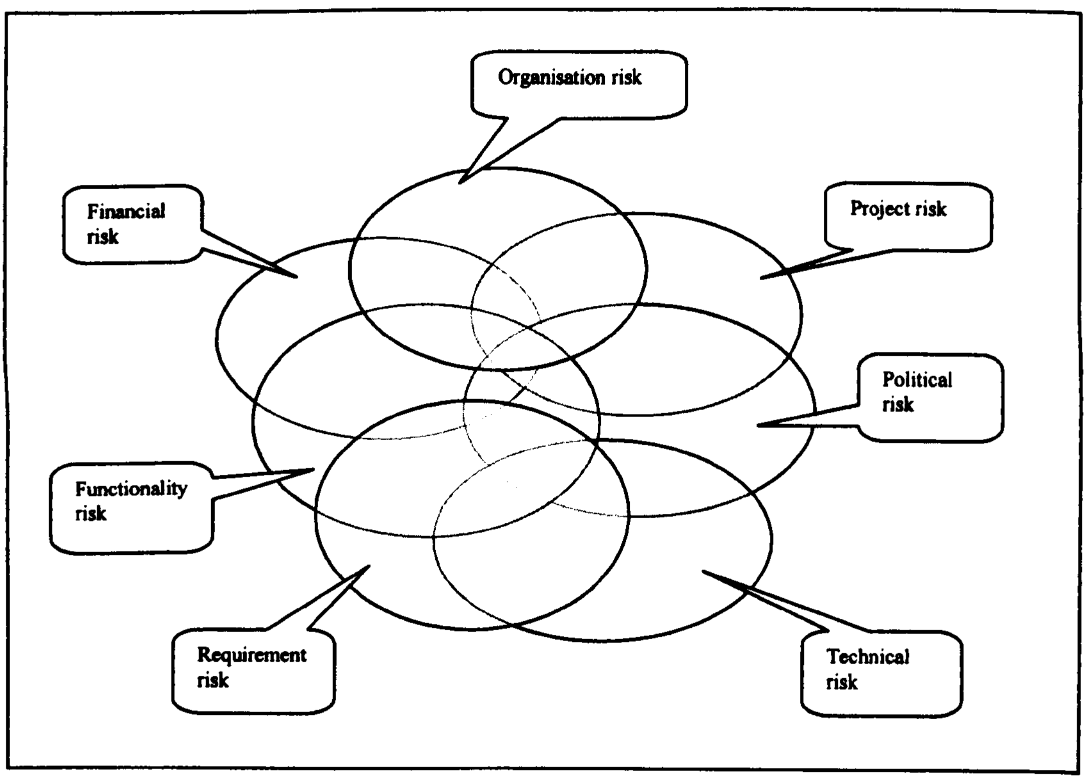


Figure 4.1 : Overlapping of risk components (adapted from; Sherer & Alter, 2004)

Based on Figure 4.1 above and the literature survey showed in Table 4.2, it can be argued that the majority of the groupings, components, ranking of risk lists produced a rather overlapping risk factors and components. The overlapping of risk factors, with the lacking of organized framework would limit the usefulness of the risk components or risk models (Alter and Sherer, 2004). Without a more robust and structured framework, it is difficult to identify the source or origin of the risk factors and tracking the risk. The lack of organized framework could make it more difficult for managers and practitioners to identify, analyse the risk or even to suggest possible risk management strategies. The overlapping of many risk factors could make the risk management processes difficult and time consuming, especially as the risk management process also involved tracking and monitoring of the risk response strategies implemented. In addition, as most of the software development lifecycle process involved a lot of iterative process within the cycle, the risk management process can be a daunting task and complex.

In order to organize the risk factors in a structured framework, the research structured the risk factors based on the Project Management principle of managing a project. This is based on the fact that Project Management perspectives and principles are generally acceptable concept and widely used by most businesses and project managers for their projects. Furthermore, risk management is one of the knowledge areas in Project Management Body of Knowledge (PMBOK).

The steady progression framework within the project management principle could be an appropriate framework in dealing with the risk factors in software projects, as the more relevant risk factors can be identified specifically for any particular cycles, phases, activities or processes, whether it is in the early phase or the later phase. As one risk may still occur in more than one phase, one might argue that, there probably still going to be an overlapping of risk factors. But, this new framework could still allow the practitioners to be able to focus on the risk in that particular stage, as the main focus is to handle and monitor the risks factors through the life cycle stages in a more systematic and structured manner. By using this proposed framework, the practitioners or business managers may also be able to justify the necessary and the relevancy of the most appropriate risk management strategies. In addition, by way of understanding through the project management lifecycle stages and its related processes and activities, additional new risk factors can still be added to the relevant stages of the lifecycle.

The literature reviews also highlighted the importance and relevancy of project management construct and issues in the risk categories, dimensions, or even the risk management strategies; within the discussion of software project failures and success (Keil et al, 2004; Baccarini et al, 2004; Cannon, 1994; Kwak & Stoddard, 2004). Since project management issues were commonly cited in the literatures, on the basis of this, the project management perspective was chosen for the research as the new framework for organising the risk factors in software project. The project management principles and perspectives were also quite easily understood and communicated by most practitioners, business managers or even other non-IT related staff within the organization. This could assist the team members to have a working understanding of risk factors within the context of their scope of work, and being more responsible and accountable for the consequences of their actions.

Furthermore, as most research suggested that risk identification and risk management process is also the responsibilities of other team members (and not just the project managers), this new framework could be a medium of enhancing communications between the team members in dealing with risk factors in software project. As pointed out by Kuruppuarachchi et al (2002); Pinto (1998); Tan (1996); and Jiang (1996), Verner et al (1999), good communications could provide the software project with a greater chance of success.

The risk factors identified from the literature survey shown in Table 4.2 were incorporated in this new framework of software development life cycle. Risk factors identified and gathered based on the researchers knowledge, understanding and informal discussions with others, were also included in the framework. The list of risk factors in Table 4.3 – Table 4.8 is not necessary an exhaustive or a complete list of risk factors, but it is a complete list of risk factors at the time of conducting the literature survey. Within this new framework, any additional or new occurrence of risk factors can still be added into the relevant stages. The new framework of software development lifecycle used to classify risk factors is shown below :-

The framework of software development lifecycle used to classify the risk factors is shown below :-

4.6.1. Feasibility study

Most projects begin with a feasibility study to determine whether the proposed new system can be implemented to generate desired output given organisational constraints. (Martin, 1992; Hussain, 1995; Clifton, 1990; Senn,1995; Taylor, 2003). The main focus is to estimate the principal costs and whether the cost of the system compares favourably with the expected benefits. The cost benefit analysis is not necessarily on financial terms only but also must be viewed from the economical, technical, organisational perspectives in order to reach into a prudent and justifiable decision making whether an IT project is feasible or not. Through the feasibility study, not only it helps to determine whether the company has the technical and resources capabilities to do the project, but perhaps more important, it gives some ideas of whether the project would contribute to the company's growth plan. The feasibility study is also important to ensure that the system fits in with the organization's current or future ways of working.

4.6.2. Project planning

The primary purpose of planning is to establish a set of directions in sufficient detail to tell the project team exactly what must be done, when it must be done, and what resources to use in order to produce the deliverables of the project successfully. (Meredith & Mantel, 2003; Martin, 1992; Alter,2002). Almost all projects, because of their relative duration and often prioritized control of resources, require formal and detailed planning. The plan must be design in a way that the project outcome also meets the objectives of the organization, as reflected by the project portfolio or other strategic selection process used to approve the project during the feasibility study stage. (Kerzner, 2006).

4.6.3. Requirement

Although requirement is not part of a generic project management life cycle, but most research in this area highlighted the important of requirements for a successful software development project. Detail requirement analysis contains a comprehensive system requirements that include detail descriptions of the system inputs and outputs, and the processes used to convert the input data into the outputs (Alter,2002). Although IS analysts are typically responsible for drafting and revising the requirements specifications, organization are also responsible for making sure that the written requirements are correct and complete. It clarifies the scope and purposed of the proposed project by describing the work processes that will be affected and how they will be performed using the system. If the requirement of the project is too narrow, the tendency is that the project may not meet its functional needs. On the other hand, if the requirement is too broad, the project may not be finished. The balanced of this is necessary to stay within the functional requirements of the system, the resources, budget and schedule stipulated. The requirements were documented accurately and in detail in order to provide a sound basis for successful systems design and development stage.

4.6.4. Development stage

The 'development stage' is being used as the terminology for this stage is to reflect this stage as where the team develops the software after the previous stages of planning and requirement stage. During the development stage, the technicality and complexity of the proposed system becoming clearer focused, as the description of what the system will do, the configuration of the system and computer environment, the compatibility with existing system and the prototype or working model being detailed together. Based on the detailed requirement analysis, the IT specialists and the design team will design the physical system. In system design, the team decides what hardware and systems software to use to operate the system, design the structure and contents of the system (Martin, 2002). This detailed design document will then be given to programmers to produce the computer programs and coding. The programmers also developed the databases and files to be used by the system. The main involvement of the users would be in assisting to interpret the requirements and design documents. The design stage creates the bridge between the user's need and the hardware and software capability. It is concerned with mapping the business need into a technical solutions and design details, which ensure that the system is reliable, secure and adequate capacity.

4.6.5. Implementation stage

After the development stage, the IT software is implemented within the organizations. IT projects frequently involve changes to the jobs of the people who will use the system, and these changes must be anticipated and well planned. Bailey (1998) studied in depth the type of change characteristics; tangible, conceptual and personal; that could be expected once a project has been implemented. There is a need for careful evaluation of planning strategies and methodologies that would facilitate these changes. The major problems in system installation usually lie in adapting the organisation to the new system that is changing how people do their work (Mobey, 2002). The conversion process may require attitudinal changes. It is probably a mistake to assume that people will change their behaviour in the desired or expected way. Installing the hardware and software can be a challenge when the new system involves technology that is new to the organisation.

4.6.6. Operation & maintenance

Successful operation of an application system requires that people and computers work together (Martin,2002). If hardware or software fails or people falter, system operation may be unsatisfactory. In any IT project, there are thousands of things that can go wrong, and most organizations operate many systems simultaneously. It takes excellent management of computer operations to make sure that everything works well consistently, and to contain and repair the damage when things do go wrong. Regardless of computer size, periodic evaluation of operations should take place ,evaluation being the comparison of actual performance with the objectives. (Hussain,1995). If performance is unsatisfactorily, either systems maintenance, minor modification, redevelopment or major change is triggered. The efficiency and effectiveness of the system once changes are made will subsequently be reviewed at scheduled evaluation sessions.

4.7. The stages IT risk

4.7.1. Feasibility study stage IT risk

The purpose of the feasibility study is to ascertain whether the desired objectives of the system can be achieved within the existing economical, financial, organizational and technological constraints (Hussain, 1995; Alter, 2002; Martin, 1992). The focused will be around the general ideas of helping management in the decision making processes to determine whether the project should be pursued. Various solutions and alternatives to the IT issues or problems are examined during the study. The focus of the feasibility study for the software project should not be too narrow focus on *the IT technical issues or resources,* but the *management and business impact issues must not be overlooked* (Willcocks & Griffiths, 1994; Addison, 2003). This also raised the question of whether the worthiness of committing organisational resources to the software project, or whether the resources might be more useful elsewhere, or even *investing in the right software project and technology* (Addison, 2003). On the contrary, the management also need to consider the cost of not pursuing the software project and perhaps losing ground on the competitors in the market (Taylor, 2003; Kingston, 2004). The organisations need to consider as to whether the firm can afford to build the system and justify their investments (Burch et al, 1983; Clifton, 1990; Laudon, 1995; Senn, 1995). As feasibility study is the early part of the stages and not much is known about the software project, it is crucial the risk factors in this stage being considered thoroughly. The risk factors in this stage can have very significant impact on the progress of the software project. The *cost benefit analysis* of the software project could shows relevant and feasible alternatives in terms of major costs that will be incurred during the development and running of the system, together with the major benefits that are expected to accrue (Kingston, 2004). Generally, the overall benefit must outweigh the overall cost, but not necessary in financial terms only. Failure in the identification and assessment of risk factors at the project feasibility stage of the software project might manifest as project problems later-on, and have an impact on project success. After the feasibility study stage, recommendations are normally made whether to proceed with the project and the next stage. Risk factors incorporated in this stage is shown in Table 4.3 below:-

Table 4.3 : Risk factors for the feasibility study stage

Stages	**Risk factors**	**Research coding**
Feasibility study	Inproper justification of cost benefit analysis and evaluation criteria from feasibility study	F1
	Too narrow focus on the technical IT issues	F2
	Overlooked the management and business impact issues	F3
	Wrong justification of investment alternatives and opportunity cost	F4
	Inappropriate technology chosen from the feasibility study	F5

4.7.2. Project planning stage IT risk

Successful software project completion requires detailed and meticulous planning, careful monitoring, *reviews and updating* of the state of the software project against *the scope and objectives* (Keil et al, 2008; Tesch et al, 2007; Perera et al, 2006; Wat & Ngai, 2005; Wallace & Keil, 2004). The management need to consider the difficulties of *estimating the resource required, planning the system conversion* and coordinating the work of staff, and ensuring that the *overall software project is completed within the project schedule* (Keil et al, 2008; Ming Han et al, 2007; Eun Hee Kim et al, 2006; Wat & Ngai, 2005; Wallace & Keil, 2004). A slippage at any particular phase, cannot always be corrected by simply putting more staff or allocate money at the problem, but a *contingency plan and a plan for change management process for the software project* could help (Helio Costa et al, 2007; Perera et al, 2006; Wat & Ngai, 2005). For a successful software project, normally a lot of time is spent on the planning phase and sometimes longer than it takes to complete the rest of the project itself. It is important to remember that the success of a software project starts with the project plan. The initial project plan needs to be as thorough and detailed as possible including the *critical and non-critical activities, the project milestones* and the *work breakdown structure* (Mark Keil et al, 2008; Ming Han et al, 2007; Helio Costa et al, 2007; Eun Hee Kim et al, 2006; Wallace & Keil, 2004) . The *project management* and *development team* also need to be established. Other risk factor that was also included in this stage is *lack of quality control procedure, unclear line of decision making authority throughout the project* and the *success criteria* for the project (Helio Costa et al, 2007; Perera et al, 2006; Wallace & Keil, 2004). Poor project planning in the software project makes problems or setbacks more likely to occur and less likely to be noticed and properly dealt with when they do occur. Without an effective project plan, it is going to be hard to know whether the software project is performing well and justify it with the project goals and deliverable output. Lack of attention to project planning stage may also escalate the software project and lead to project failure.

Table 4.4 : Risk factors for the project planning stage

Stages	Risk factors	Research coding
Project planning	Unclear project scope + objectives	P1
	Undefined project success criteria	P2
	Lack of quality control procedure and mechanism	P3
	Project milestones for stages not well establish	P4
	Improper change management planning	P5
	Inaccurate estimate of resources	P6
	Unrealistic project schedule	P7
	Inadequate detail breakdown structure	P8
	Critical and non-critical activities of project not identified	P9
	Project management and development team not properly set up	P10
	Unclear line of decision making authority throughout the project	P11
	Lack of contingency plan/back up	P12
	System conversion method not well planned	P13
	Improper planning of timeframe for project reviews and updating	P14

4.7.3. Requirement stage IT risk

Although requirements are not very well mentioned in the literature as part of project management life cycle, but requirements were highlighted by most researches in IT risk as one of the main categories or dimensions. This can also be seen in the list of risk factors in Table 1. Even, a number of researchers had also classified requirements as one of the main category or dimensions. The essence of the stage is to achieve a thorough and insightful understanding of the requirements of the system (Hussain, 1995; Senn, 1995; Clifton, 1990). As a result of this, a detailed systems requirements and specifications is materialized that will be use as a basis for development stage. Poor requirements or failure to define accurate requirements can lead constant changes of requirements and the creation of a system that does not fit the users' needs, thus resulting in project failure. The important risk factors in this stage include the *clarity of the requirements* (Boehm, 1991; Schmidt et al, 2001; Perera et al, 2006; Eun Hee Kim et al, 2006), *adequacy of the requirements, any unnecessary requirements* (Eun Hee Kim et al, 2006; Tesch et al, 2007), *validations of the requirements* (Schmidt et al, 2001; Helio Costa et al, 2007) *and also users involvement in the requirements stage* (Wallace et al, 2004; Ropponen & Lyytinen, 2000; Perera et al, 2006). Table 4.5 show the risk factors for the requirement stage included for this study.

Table 4.5 : risk factors for the requirement stage.

Stages	**Risk factors**	**Research coding**
Requirement	Unclear and inadequate identification of systems requirements	R1
	Incorrect systems requirements	R2
	Misinterpretations of the systems requirements	R3
	Conflicting system requirements	R4
	Gold platting or unnecessary functions and requirements	R5
	Inadequate validation of the requirements	R6
	Lack of users involvement in requirement stage	R7

4.7.4. Development stage IT risk

Development stage is the transformation of a general system requirement into hardware and software that accomplish the required functions (Hussain, 1995; Alter, 2002). This is the stage where all the necessary groundwork and investigations did during the feasibility stage, project planning stage and requirement stage deemed very crucial and important. Based on the requirements and project plan from the earlier stages, the development stage is about specifying how the new system is to achieve the functions, outputs and also operational performance (Taylor, 2003; Alter, 2002). In this stage the risk factors of the software project included were predominantly *technological related factors* such as the *development methodology used, the technical complexities of the technology and the testing of the unit or modules* (Barki et al, 1993; Keil et al, 1998; Jiang & Klien, 1999; Ropponen & Lyytinen, 2000; Cule et al, 2000; Wallace et al, 2004; Wat & Ngai, 2005; Tesch et al, 2007). *Human related risk factors* such as the *experienced and skills of the project manager and development team, staff resources issues, communication factors and also users involvement* (Willcocks & Griffith, 1994; Jiang & Klein, 1999; Schmidt et al, 2001; Wallace & Keil, 2004;Helio Costa et al, 2007; Ming Han et al, 2007), were also included in the development stage. Table 4.6 shows the risk factors for the development stage in this research.

Table 4.6 : Risk factors for the development stage.

Stages	**Risk factors**	**Research coding**
Development	Inproper handover from the requirement team	D1
	Inappropriate development methodology used	D2
	Unsuitable working model and prototype	D3
	Programming language and CASE tool selected not adequate	D4
	High level of technical complexities	D5
	Project involves the use of new technology	D6
	Difficulty in defining the input and output of system	D7
	Immature technology	D8
	Technological advancements and changes	D9
	Failures and inconsistencies of unit/modules test results	D10
	Failure of user acceptance test	D11
	Time consuming for testing	D12
	Resources shifted from project due to organisational priorities	D13
	Changes in management of organisation during development	D14
	Lack of users involvement and commitment	D15
	Team members lack specialized skills required for the project	D16
	Ineffective communication within development team members	D17
	Ineffective communication between users and development team members	D18
	Inadequately trained development team members	D19
	Team members not familiar with the tasks/processes being developed	D20
	Inexperienced team members	D21
	Lack of commitment to project among development team members	D22
	Ineffective and inexperienced project manager	D23
	Frequent staff turnover within project team	D24
	Conflicts between users and development team members	D25
	Conflict among users	D26
	Conflicts within development team members	D27
	Excessive schedule pressure and overworked	D28
	Lack of control and coordination within the project	D29
	Overreliance on subcontractor or vendors/suppliers	D30
	Redundancies and overlapping of activities/processes	D31
	Lack of regular reviews against goals	D32
	Large project size	D33
	Tracking of problems within the processes/activities	D34
	Improper sequential of processes/activities	D35

4.7.5. Implementation stage risk

It is a process of putting the system into operation in an organization after the development stage (Alter, 2002; Palisha, 2002; Martin, 2002). The process may involve substantial changes to the people, system, organisation and also working processes. These changes may be minimal or even drastic and can cause intra-organisational issues and tensions. The main issue here is whether the IT system can be converted as effectively and systematically from the old system to the new system (Mobey, 2002). It is difficult to visualise from project specification or design, how the IT system will work or have impact on the organisation. Whilst good design of system is important, successful change requires implementation planning, execution and improvisation to deal with resistance and unforeseen events (Lynne & Benjamin, 1997). The implementation stage may also dependable on how good is the development process previously. The risk factors of the software project in this stage need to be considered carefully as any difficulties and changes arising from the risk factors may incur considerable losses in terms of times, financial and other resources spent. Risks factors related to *conversions of the system, disruption to existing processes, users expectation and adaptability to new system* were very important (Willcocks & Griffiths, 1994; Jiang & Klien, 1999; Cule et al, 2000; Wallace & Keil, 2004; Perera et al, 2006). Other not directly technical related factors such as *communications, training, documentation and number of units involved during implementation* must not be overlooked (Mark Keil et al, 2008; Tesch et al, 2007; Ming Han et al, 2007; Wallace & Keil, 2004). *Users' involvement* is also crucial as, they are the main user or recipient of the new system or software (Willcocks & Griffith, 1994; Jiang & Klien, 2001; Schmidt et al, 2001; Addison, 2003; Ming Han et al, 2007). Table 4.7 shows the risk factors included in the implementation stage for the study.

Table 4.7 : Risk factors for the implementation stage

Stages	**Risk factors**	**Research coding**
Implementation	Unsuitable conversion/installation method	IM1
	Loss of data during conversion/installation	IM2
	System failure during conversion/installation	IM3
	Loss of performance during installation	IM4
	Improper implementation sequence modules/activities	IM5
	Disruption to existing operation/processes	IM6
	Difficulty in configuration of system and computer environment/platform	IM7
	Time constraints in implementation	IM8
	Large number of interfaces to other system required	IM9
	Users adaptability to new system	IM10
	Users lack understanding of system capabilities and limitations	IM11
	Failure to manage end-user expectations	IM12
	User resistance to change	IM13
	Feedback from users not properly analyzed	IM14
	Time constraints of training	IM15
	Outlining training schedule	IM16
	Lack of knowledge and experience of system administrator/configuration manager	IM17
	Lack of knowledge and experience of implementation team	IM18
	Ineffective communication between users and implementation team members	IM19
	Ineffective communication within implementation team members	IM20
	Changes in management of organisation during implementation	IM21
	Resources shifted from project due to organisational priorities	IM22
	Projects affects large number of user departments/units	IM23
	Inadequate documentation for implementation	IM24

4.7.6. Operation maintenance stage IT risk

The way in which hardware, software, the database and computer personnel are deployed will determine whether operations are optimised (Hussain, 1995; Martin, 2002). Successful operation of an application system requires that people and computers work together. If hardware or software fails or people falter, system operation may be unsatisfactory. As users and managers gain experience with IT systems, they may become more aware of the potentialities of computer processing and place increased demands on systems in existence. Organisations that want to stay at the forefront of technology will find that their IT systems need *frequent modifications and improvements*, or even *stay ahead their competitors*. It is very important to have the *support of the management throughout the life of the software*, to stay competitive (Addison, 2003; Wallace & Keil, 2004; Tai Hui & Biau Liu, 2004; Wat & Ngai, 2005). Maintenance of the software normally refers to the process of making changes to a system after it has been put into production mode or operation up and running. The most obvious reason for maintenance is to correct errors in the software and hardware that were not discovered and corrected prior to its initial implementation. Usually a number of *bugs or viruses* in a system do elude the testing process and for a large or complex IT system, it may take several months or even years to discover (Wat & Ngai, 2005; Perera et al, 2006); . Maintenance of the software may also be required to adapt the system to *changes in the environment*, *the organization*, other systems, *new hardware and systems software*, and *government regulations* (Barki et al, 1993; Willcocks & Griffith, 1994; Cule et al, 2000; Schmidt et al, 2001; Addison, 2003;Wat & Ngai, 2005). Table 4.8 show the risk factors included in the operation maintenance stage for the study.

Table 4.8 : Risk factors for the operation maintenance stage

Stages	Risk factors	Research coding
Operation & maintenance	Lack of organisation's commitment throughout project life	OP1
	Systems not performing accurately and effectively	OP2
	System failure and breakdown	OP3
	Inconsistencies of output produced	OP4
	Inadequate user documentation	OP5
	Poor maintenance schedule	OP6
	Poor maintenance procedure	OP7
	Lack of technical support	OP8
	Threat of hackers	OP9
	Viruses/bugs	OP10
	Unauthorised user/sabotaj/abuse	OP11
	Inadequate safety/security features	OP12
	Changes in market condition and organisation priorities	OP13
	Systems and programming languages become obsolete	OP14
	Actions taken by competitors	OP15
	Software not flexible in supporting new requirements and changing user needs	OP16
	Cost of training	OP17
	Lack of continuous IT investment to sustain competitiveness	OP18
	Price fluctuations of hardware and software	OP19

4.8. Summary

The chapter explain the previous research perspectives and the reasons in the proposition of a new framework for the software development life cycle. The chapter shows how the project management framework of life cycle can be used to organize the risk factors of software project from the literatures. The extraction of the risk factors form the literature was explained and risk factors were extracted for the proposed new framework. The risk factors were incorporated into the relevant stages of the life cycle. This new framework of risk in software project will be used in the questionnaire design for the data collection process. The framework will be used to survey and analyse the opinions and perceptions of the software practitioners in relation to the likelihood occurrence of risk factors and their impact on the cost overrun of a software project.

CHAPTER 5 DESCRIPTIVE STATISTICS & DATA RANKING

5.1. Introduction

In this survey, a questionnaire was developed and distributed among practitioners in different level of information systems professions, involved in software development projects. The data obtained were grouped, organised in order to be analysed and discussed in the next section. Overall findings of the questionnaires are presented in this chapter. This chapter only explain the descriptive statistics of the findings. Further analysis and discussions of the results and the perceptions of the practitioners will be explained in detail during the later chapters.

The first section of the questionnaire (Section A) includes 4 questions relevant to the respondents' general information. Section A asked the respondents the nature of their companies, respondent's designation in that company and their experiences in software development project in terms of number of projects undertaken and years of involvement.

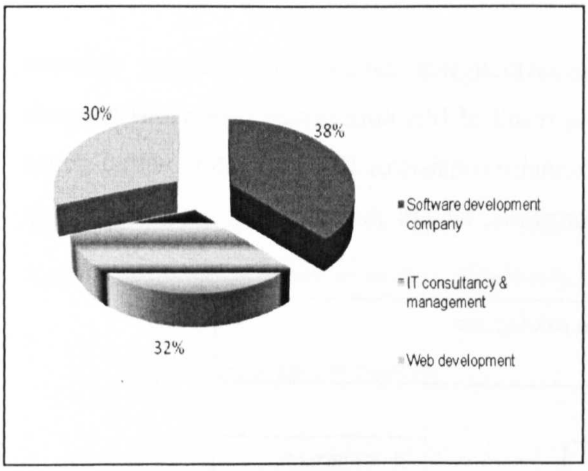
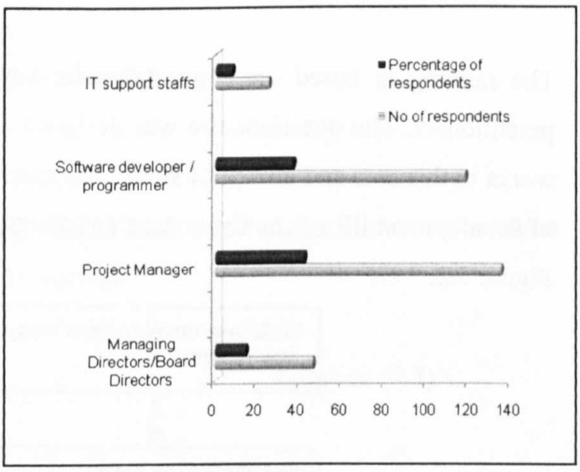


Figure 5.1 : Companies profile



Graph 5.1 : Respondents job description

The findings showed 46 respondents (14%) were Managing Directors/Board of Directors of companies, 135 respondents (42%) were Project Managers, 118 respondents (36%) were Software developers/programmers, and 25 respondents (8%) were IT support staffs of their respective companies. The companies consist of 122 (38%) Software development companies, 104 (32%) IT Consultancy and management companies and 98 (30%) Web development companies.

5.2. Data ranking

Ranking is based on list of risk factors being ranked on their importance as a result of the rating by the respondents. The need for ranking normally applies when there were huge of set of data and the need to find and select similar indicators or common themes and trends for the research.

This chapter examines the statistical techniques used to rank the data obtained from the questionnaire survey, which consists of 104 risk factors within 6 stages. In this study, the SPSS and Microsoft Excel were used for the ranking analysis. The method of evaluation and ranking is based on statistical analysis such as (Field, 2005; Morgan et al, 2004; Punch, 2006) :-

- The average weighted mean
- Standard deviation
- Coefficient of variation
    - The ratio of standard deviation as a percentage (%) of the mean.
    - For comparing the relative variability of various responses.
    - The lower variation coefficient, the better is the variability.
- Severity index
    - Ranking of the indicators according to their significance.
    - The higher percentage (%), more significance is the factor.

The ranking is based on a questionnaire survey which was commissioned among software practitioners. The questionnaire was designed as a result of literature reviews, previous research works in this area and also pilot study. The questionnaire consists of 104 risk factors with 6 stages of development life cycle. Each stage has a number of risk factors attributed to it. This is shown in Figure 5.2.

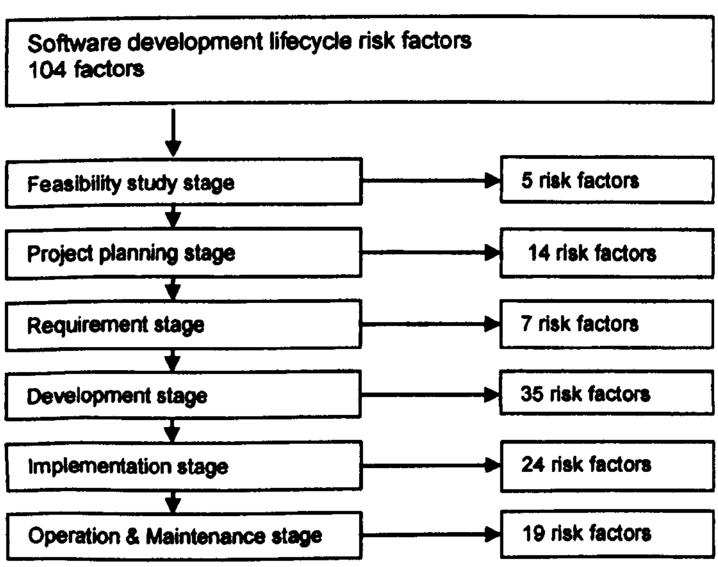


Figure 5.2. Risk factors questionnaire structure

5.3. Analysis and ranking

A mean weighted rating for each risk factor is computed to indicate the importance of each indicator, using the equation 4.1 below.

Mean weighted rating = **[$\sum$ (R*F)] / n** equation (5.1)

Where;

R = rating of each risk factor (1,2,3,4,5)

F = frequency of responses

n = total number of responses (n = 324)

Severity index (S.I) measure is to rank the indicators according to their significance. Equation 5.2 presents how S.I is calculated :-

S.I. = { [$\sum$(W*F)] / n } * 100 % equation (5.2)

Where;

W = weight of each rating (1/5, 2/5, 3/5, 4/5, 5/5)

F = frequency of responses

n = total number of responses (n = 324)

The ratio of standard deviation (SD) as a percentage of the mean, is called Coefficient of variation (COV) and is for comparing relative variability of responses.

COV = (S / M) * 100 % equation (5.3)

Where;

S = standard deviation

M = weighted mean sample

List of risks, derived from the literatures was provided to the respondents who were asked to rate each risk in terms of the likelihood occurrence of the risks using the Likert scale of 1-5 (1- none; 2 unlikely; 3-likely; 4-highly likely; 5-very highly likely). The respondents were also asked to rate each risk factor impact on the cost overrun of a software project using the scale below :-

a. Very low (1-10% overrun)
b. Low (11-20% overrun)
c. Moderate (21-30% overrun)
d. High (31-40% overrun)
e. Very High (> 40% overrun)

5.4. Rating and ranking of Likelihood occurrence of risk factors

For the purpose of this chapter, the full Table 5.1 illustrating the statistical ranking results for all 104 indicators is shown in the Appendixes. In the Table 5.1, the overall ranking, the Kendall ranks and the ranking by each practitioner for every risk factor are presented.

From the Table 5.1, the average weighted mean for the risk factors varies from 1.45 to 4.44, with the overall mean of 2.86. The severity indices range within 29 % to 88 %. As it can also be seen from Table 5.1, the top 30 ranked risk factors were dominated by the indicators from the Project Planning stage and Development stage, where the highest ranked factor was (P6-inaccurate estimate of resources) with a mean of 4.44 and severity indices of 88.7 %. An overall examination of the first 30 ranked risk factors in Table 5.1 indicates that all first 30 ranked factors have a minimum mean value of 3.37 (which is higher than the overall mean of 2.86) and severity indices of 67.35 %. This means that the first 30 ranked risk factors seem to be important as viewed by the respondents.

The overall ranking for risk factors (P6–inaccurate estimate of resources) is 1st out of 104, the Managing Directors/Board of Directors and the Developer/Programmer also rated P6 as the highest ranked. The Project Manager ranked it 3rd out of 104, and the IT support staffs ranked it 2nd out 104. This factor carries a severity index of 88.7%, a coefficient of variation of 19.23%, standard deviation of 0.854 and average weighted mean of 4.44.

5.4.1. Feasibility study stage risk factors

Feasibility study stage consists of 5 risk factors. Ranking results in Table 5.2 shows that there are 2 factors (F1, F3) with ranking among the first 30 ranked indicators. In Table 5.3, factor F1 is considered as the highest ranked indicators for the Feasibility stage, with the mean of 3.6 and severity index of 72.04%. It has an overall ranking of 21st out of 104; Managing Directors/Board of Directors ranked 5th out of 104; Project Managers ranked 23rd out of 104; Developer ranked 29th out of 104 and IT support staffs ranked 33rd out of 104.

Table 5.2 : Ranking of likelihood occurrence for Feasibility stage

			Ranking							
Ref	Mean	Standard deviation	Managing Directors	Project Manager	Developer	IT staff	Coefficient of variation	Severity index	Kendall Mean ranking	Overall Ranking
F1	3.60	0.94	5	23	29	33	26.11	72.04	71.97	21
F2	3.34	0.871	12	43	26	61	26.07	66.85	67.22	32
F3	3.47	1.06	31	15	35	45	30.54	69.44	67.54	24
F4	2.28	0.835	68	66	69	64	36.6	45.62	45.03	67
F5	2.44	0.862	60	60	62	56	35.32	48.83	50.39	61

For factor F3, it has an overall ranking of 24th out of 104; Managing Directors/Board of Directors ranked 31st out of 104; Project Managers ranked 15th out of 104; Developers ranked 35th out of 104 and IT support staffs ranked 45th out of 104. Factor F3 has a mean of 3.47 and severity indices of 69.44%.

5.4.2. Project Planning stage risk factors

Table 5.3 : Ranking of likelihood occurrence for Project Planning stage

			ranking							
Ref	Mean	Standard deviation	Managing Directors	Project Manager	Developer	IT staff	Coefficient of variation	Severity index	Kendall Mean ranking	Overall Ranking
P1	4.17	1.07	6	6	5	4	25.65	83.4	84.20	4
P2	3.69	0.829	20	19	18	17	22.46	73.7	77.24	19
P3	2.72	0.964	45	50	53	40	35.44	54.32	54.30	51
P4	2.41	0.826	57	63	61	48	34.27	48.27	44.18	62
P5	3.56	0.976	24	22	25	26	27.41	71.17	70.48	22
P6	4.44	0.854	1	3	1	2	19.23	88.7	91.84	1
P7	4.41	0.784	2	2	2	3	17.77	88.21	92.18	2
P8	2.56	0.986	51	56	56	44	38.51	51.11	48.95	55
P9	3.73	1.019	11	12	22	55	27.31	74.69	75.48	16
P10	1.81	1.107	88	89	89	85	61.16	36.17	28.69	89
P11	3.99	0.897	21	1	23	22	22.48	79.88	82.15	9
P12	3.66	0.615	13	21	21	21	16.80	73.27	73.79	20
P13	3.30	0.995	25	28	34	72	30.15	66.05	63.67	34
P14	3.19	0.830	36	33	40	47	26.01	63.89	64.35	41

In the Project Planning stage, 8 risk factors from the 14 factors in this stage were ranked in the first 30 highest indicators namely; P1, P2, P5, P6, P7, P9, P11, P12. This means that more than 50% of the indicators in the Project Planning stage were ranked in the first 30 highest indicators These factors' means range from 3.56 to 4.44. Also their severity indices vary from 71.17% to 88.7%. The score of average weighted mean and the severity indices for all of these indicators are very high in comparison with other stages. Factor P6 with a mean of 4.44 and severity indices of 88.7 %, is considered is the highest ranked indicator for this stage. Apart from factor P6, two more factors, P7 and P1, have an overall ranking of 2nd and 4th (out of 104) respectively. Factor P7 has a mean of 4.41 and severity indices of 88.21%, whereas, factor P1 has a mean of 4.17 and severity indices of 83.4%. Both of these 2 factors were ranked in the top 6 (out of 104) by Managing Directors/Board of Directors, Project Managers, Developers and IT support staffs, with a low coefficient of variation of 17.7% (P7) and 25.6% (P1).

5.4.3. Requirement stage risk factors

Table 5.4 : Ranking of likelihood occurrence for Requirement stage

			ranking							
Ref	Mean	Standard deviation	Managing Directors	Project Manager	Developer	IT staff	Coefficient variation	Severity index	Kendall Mean ranking	Overall Ranking
R1	4.08	0.767	10	9	8	7	18.79	81.6	84.02	7
R2	2.15	1.32	73	71	74	70	61.39	42.96	37.53	73
R3	3.32	0.852	23	26	36	66	25.66	66.36	68.85	33
R4	2.44	0.986	64	57	64	59	40.41	48.89	49.92	60
R5	1.90	1.040	87	80	86	93	54.73	37.96	34.05	85
R6	3.82	0.974	42	11	11	10	25.49	76.36	77.60	12
R7	3.41	1.658	50	24	24	37	48.6	68.21	64.75	26

For the Requirement stage, 3 factors from the 7 factors in this stage were ranked in the 30 highest ranked indicators. They were R1, R6 and R7. The factor R1 has an overall rank of 7th out of 104, and has a mean on 4.08 and severity indices of 81.6%. In fact, with a low coefficient of variation of 18.7%, factor R1 was ranked in the top10 (of 104) by all 4 categories of respondents. The indicator R6 has an overall rank of 12th out of 104; Managing Directors/Board of Directors ranked 42nd of 104; Project Managers ranked 11th; Developers ranked 11th and IT support staffs ranked 10th out of 104. It has an overall mean of 3.82 and severity indices of 76.36%. The risk factor R7 being ranked 50th of 104 by Managing Directors/Board of Directors; 24th out of 104 by Project Managers; 24th of 104 by Developers and 37th out of 104 by IT support staffs. It has an overall ranking of 26th, with the weighted mean of 3.41 and severity indices of 68.21%.

5.4.4. Development stage risk factors

In the Development stage, 11th of the 35 factors in that stage being ranked in the first 30 highest ranked risk factors. These factors have an overall mean in the range of 3.37 to 4.3. The severity indices are in the range of 67.35% - 85.93%. Factor D17 seemed to be the most important risk factor for this stage and has an overall rank of 3rd out of 104 indicators, and the severity indices of 85.93%. The 4 group of practitioners also ranked factor D17 as their top 5 highest ranked risk factor, with a low coefficient of variation of 20.93%.

Table 5.5 : Ranking of likelihood occurrence for Development stage

			Ranking							
Ref	Mean	Standard deviation	Managing Directors	Project Manager	Developer	IT staff	Coefficient of variation	Severity index	Kendall Mean ranking	Overall Ranking
D1	3.14	1.056	30	44	47	27	33.6	62.72	63.72	43
D2	4.16	0.854	18	4	3	9	20.52	83.21	86.57	5
D3	2.53	0.830	52	58	57	46	32.8	50.62	47.52	56
D4	2.51	1.139	53	59	58	51	45.37	50.25	46.79	57
D5	3.79	0.944	48	13	9	13	24.9	75.8	78.36	13
D6	2.34	1.036	66	62	66	71	44.27	46.79	43.81	65
D7	3.09	1.175	28	41	45	75	38.02	61.73	61.54	45
D8	2.22	0.887	70	67	49	73	39.95	44.44	40.61	70
D9	2.06	0.819	76	77	76	78	39.75	41.23	35.25	76
D10	2.25	0.975	67	72	68	62	43.33	45.06	38.69	68
D11	3.21	0.937	38	36	38	35	29.19	64.2	62.27	40
D12	3.69	0.849	15	20	20	18	23.01	73.89	77.11	18
D13	3.35	1.047	4	38	43	29	31.25	67.04	64.45	31
D14	1.97	1.007	77	87	78	67	51.11	39.38	31.04	79
D15	3.92	1.139	9	14	14	8	29.05	78.46	77.78	11
D16	2.48	1.176	58	54	60	68	47.41	49.63	48.57	59
D17	4.3	0.9	3	5	4	1	20.93	85.93	88.01	3
D18	3.37	0.982	29	29	32	30	29.13	67.36	66.24	30
D19	1.96	1.052	78	83	79	79	53.67	39.14	30.99	81
D20	1.92	0.725	83	84	84	83	37.76	38.4	29.88	84
D21	2.15	1.179	74	69	72	80	54.83	42.96	40.38	72
D22	2.90	1.036	39	48	50	34	35.72	57.9	55.62	49
D23	4.02	0.721	14	8	10	12	17.93	80.37	82.59	8
D24	3.38	0.973	34	46	13	19	28.78	67.53	67.40	28
D25	2.13	0.925	72	76	75	57	43.42	42.59	35.99	74
D26	1.75	1.048	91	95	93	84	59.88	35	28.60	91
D27	1.60	0.973	99	99	100	102	60.81	31.91	23.19	100
D28	3.98	0.924	40	7	6	6	23.21	79.63	80.62	10
D29	3.79	0.996	17	16	17	15	26.27	75.74	75.03	14
D30	1.61	0.971	100	97	99	99	60.31	32.28	26.40	99
D31	2.05	0.925	79	75	77	74	45.12	41.05	36.39	77
D32	4.09	0.806	8	10	7	5	19.70	81.73	83.43	6
D33	2.25	0.964	69	70	70	52	42.84	44.94	39.33	69
D34	1.52	0.500	101	101	101	100	32.89	30.43	22.15	101
D35	1.98	0.862	80	79	82	77	43.53	39.63	34.30	78

5.4.5. Implementation stage risk factors

As for the implementation stage, only 6 of the 24 factors in this stage were in the first 30 highest ranked indicators namely; IM9, IM15, IM19, IM22, IM23, IM24. The overall mean for these factors were in the range of 3.37 to 3.79. The range for the severity indices was 67.41% to 75.74%. All these 6 factors have an overall weighted mean of rank 15th to rank 29th. The factor IM15 was the most important risk factor in this stage with the overall rank 15th out of 104.

Table 5.6 : Ranking of likelihood occurrence for Implementation stage

			Ranking							
Ref	Mean	Standard deviation	Managing Directors	Project Manager	Developer	IT staff	Coefficient of variation	Severity Index	Kendall Mean ranking	Overall Ranking
IM1	1.94	0.744	81	81	80	81	38.35	38.89	31.91	82
IM2	2.30	0.940	65	68	67	53	40.86	45.93	41.90	66
IM3	1.78	0.819	88	92	90	89	46.01	35.68	26.42	90
IM4	2.10	0.841	75	74	73	76	40.04	42.1	37.27	75
IM5	1.67	0.851	92	100	97	94	50.95	33.33	24.13	98
IM6	1.69	0.658	93	98	96	86	38.93	33.7	24.26	96
IM7	1.94	0.744	82	82	81	82	38.35	38.89	31.91	83
IM8	3.00	0.946	56	42	44	41	31.53	60.06	57.43	47
IM9	3.37	1.016	35	47	12	14	30.14	67.41	67.56	29
IM10	1.88	1.015	84	88	85	90	53.98	37.53	27.59	86
IM11	2.37	0.998	61	65	65	60	42.10	47.41	42.03	64
IM12	3.19	1.553	54	32	31	63	48.68	63.77	57.19	42
IM13	2.70	1.039	43	52	52	42	38.48	53.95	48.68	52
IM14	3.22	0.942	33	34	37	39	29.25	64.32	61.20	37
IM15	3.79	0.841	16	17	15	16	22.18	75.74	78.29	15
IM16	3.13	1.312	63	39	41	23	41.91	62.53	57.72	44
IM17	2.39	0.823	62	64	63	54	34.43	47.78	42.16	63
IM18	2.51	0.962	55	61	59	49	38.32	50.12	45.36	58
IM19	3.46	0.958	26	27	28	28	27.68	69.26	67.97	25
IM20	3.07	0.918	44	40	42	50	29.90	61.48	57.46	46
IM21	2.81	1.209	41	51	51	31	43.02	56.17	51.44	50
IM22	3.48	0.871	27	25	27	24	25.02	69.63	68.63	23
IM23	3.71	1.015	19	18	19	20	27.35	74.2	76.04	17
IM24	3.25	1.044	46	49	16	11	32.12	64.94	65.56	35

5.4.6. Operation & Maintenance risk factors

For the Operation and Maintenance stage, only 1 factor (OP13) was ranked in the highest 30 ranked risk factor. It has an overall rank of 27th, with the mean on 3.39 and severity indices of 67.72%. Only Managing Directors/Board of Directors ranked this factor in the top 10 out of the 104 risk factors. Other practitioners ranked this factor lower in the rank; Project Manager ranked 31st out of 104; Developer 33rd out of 104; and IT support staffs ranked the factor 43rd out of 104.

Table 5.7 : Ranking of likelihood occurrence for Operation Maintenance stage.

			Ranking							
Ref	Mean	Standard deviation	Managing Directors	Project Mgr	Developer	IT staff	Coefficient of variation	Severity Index	Kendall Mean ranking	Overall ranking
OP1	3.21	1.102	22	35	39	69	34.33	64.26	60.81	39
OP2	2.20	0.618	71	73	71	65	28.09	43.95	39.72	71
OP3	1.75	0.715	94	91	92	91	40.85	34.94	28.75	93
OP4	1.73	0.695	95	93	94	96	40.17	34.57	28.76	94
OP5	2.62	0.945	49	55	55	38	36.06	52.35	47.79	54
OP6	3.22	1.429	59	30	30	58	44.37	64.38	59.29	36
OP7	2.66	0.891	47	53	54	36	33.49	53.21	51.00	53
OP8	1.73	0.695	96	94	95	97	40.17	34.57	28.76	95
OP9	1.45	0.730	102	103	103	103	50.34	29.01	20.98	104
OP10	1.45	0.730	103	104	104	104	50.34	29.01	20.98	103
OP11	1.47	0.500	104	102	102	101	34.01	29.38	21.11	102
OP12	1.68	0.710	97	96	98	98	42.26	33.52	24.83	97
OP13	3.39	1.036	7	31	33	43	30.56	67.72	64.86	27
OP14	1.85	0.807	89	86	88	95	43.62	36.91	32.10	88
OP15	1.86	0.822	90	85	87	87	44.19	37.28	32.09	87
OP16	2.99	1.026	37	45	48	32	34.31	2.623	56.58	48
OP17	1.96	1.013	85	78	83	92	51.68	39.26	35.30	80
OP18	3.22	0.627	32	37	46	25	19.47	64.32	64.65	38
OP19	1.75	0.892	98	90	91	88	50.97	35	31.41	92

5.5. Rating and ranking of Impact of factors on cost overrun

The full Table 5.8 illustrating the statistical ranking results for all 104 indicators are shown in the Appendixes. In the Table 5.8, the overall ranking, the Kendall ranks and the ranking by each practitioner for every risk factor are presented.

5.5.1. Feasibility study stage risk factors

Table 5.9 : Ranking of impact of cost overrun for Feasibility study stage.

			Ranking							
Ref	Mean	Standard deviation	Managing Directors	Project Manager	Developer	IT staff	Coefficient of variation	Severity Index	Kendall ranking	Overall Ranking
F1	3.24	1.208	17	38	41	21	37.28	64.81	65.47	33
F2	2.58	0.867	46	48	48	39	33.60	51.67	51.59	48
F3	3.44	0.995	27	22	20	25	28.92	68.7	69.97	20
F4	2.29	0.977	59	58	65	56	42.66	45.74	46.15	62
F5	2.02	0.710	71	78	84	71	35.14	40.31	39.65	77

For impact of risk factor on cost overrun, ranking results in Table 5.9 shows that only F3 was ranked among the first 30 ranked indicators by all categories of respondents. In Table 5.9, factor F3 is considered as the highest ranked indicators for the Feasibility stage, with the mean of 3.44 and severity index of 68.7%. It has an overall ranking of 20th out of 104; Managing Directors/Board of Directors ranked 27th out of 104; Project Managers ranked 22nd out of 104; Developer ranked 20th out of 104 and IT support staffs ranked 25th out of 104. For factor F1, it has an overall ranking of 33rd out of 104 and has a mean of 3.24 and severity indices of 64.81%. Both Managing Directors/Board of Directors and IT support staffs ranked F1, 17th and 21st out of 104, respectively. But, Project Manager and Developer only ranked factor F1 at 38th and 41st out of 104.

5.5.2. Project planning stage risk factors

For the impact of risk factors on cost overrun, in the Project Planning stage, 8 risk factors from the 14 factors in this stage were ranked in the first 30 highest indicators namely; P1, P2, P5, P6, P7, P9, P12, P14. This means that more than 50% of the indicators in the Project Planning stage were ranked in the first 30 highest indicators These factors' means range from 3.27 to 4.33. Also their severity indices vary from 65.49% to 86.6%. The score of average weighted mean and the severity indices for all of these indicators are very high in comparison with other stages. Factor P1 with a mean of 4.33 and severity indices of 86.6%, is considered is the highest ranked indicator for this stage. Apart from factor P1, two more factors, P6 and P7, have an overall ranking of 3rd and 2nd (out of 104) respectively. Factor P6 has a mean of 4.24 and severity indices of 84.88%, whereas, factor P7 has a mean of 4.32 and severity indices of 86.42%. Both of these 2 factors were ranked in the top 6 (out of 104) by Managing Directors/Board of Directors, Project Managers, Developers and IT support staffs, with a low coefficient of variation of 20.89% (P6) and 17.96% (P7).

Table 5.10 : Ranking of impact of cost overrun for Project Planning stage.

			Ranking							
Ref	Mean	Standard deviation	Managing Directors	Project Manager	Developer	IT staff	Coefficient of variation	Severity index	Kendall ranking	Overall Ranking
P1	4.33	0.572	3	2	2	2	13.21	86.6	91.72	1
P2	3.67	0.809	12	16	13	15	22.04	73.46	78.71	13
P3	2.56	0.854	44	49	51	41	33.35	51.3	51.50	49
P4	3.18	1.035	18	45	19	20	32.54	63.64	66.38	38
P5	3.82	0.818	4	4	24	28	21.41	76.36	81.75	12
P6	4.24	0.886	5	6	1	1	20.89	84.88	89.26	3
P7	4.32	0.776	1	1	4	5	17.96	86.42	91.16	2
P8	2.29	0.978	61	59	58	54	42.70	45.8	42.85	59
P9	3.53	0.884	16	18	17	18	25.04	70.68	75.24	19
P10	1.97	1.055	76	82	86	80	53.56	39.38	33.88	81
P11	3.04	1.131	47	41	39	53	37.20	60.86	63.14	43
P12	4.21	1.019	2	3	3	4	24.20	84.14	86.57	4
P13	2.10	0.783	69	72	74	67	37.28	41.91	38.25	71
P14	3.27	1.516	33	30	37	24	46.36	65.49	65.76	29

5.5.3. Requirement stage risk factors

Table 5.11 : Ranking of impact of cost overrun for Requirement stage.

			Ranking							
Ref	Mean	Standard deviation	Managing Directors	Project Manager	Developer	IT staff	Coefficient of variation	Severity index	Kendall ranking	Overall Ranking
R1	3.99	0.867	8	7	6	6	21.72	79.81	83.38	6
R2	2.38	0.980	62	55	56	60	41.17	47.59	50.39	56
R3	3.89	0.827	9	9	9	11	21.25	77.72	81.68	9
R4	1.98	0.684	85	79	81	75	34.54	39.69	38.65	80
R5	1.91	0.899	86	87	90	77	47.06	38.15	37.23	88
R6	3.84	1.456	6	12	11	7	37.91	76.86	76.05	11
R7	3.22	0.998	30	31	32	64	30.99	64.38	65.54	34

For the Requirement stage, 3 factors from the 7 factors in this stage were ranked in the 30 highest ranked indicators. They were R1, R3 and R6. The factor R1 has an overall rank of 6th out of 104, and has a mean on 3.99 and severity indices of 79.81%. In fact, with a low coefficient of variation of 21.72%, factor R1 was ranked in the top10 (of 104) by all 4 categories of respondents. The indicator R3 has an overall rank of 9th out of 104; Managing Directors/Board of Directors ranked 9th of 104; Project Managers ranked 9th; Developers ranked 9th and IT support staffs ranked 11th out of 104. It has an overall mean of 3.89 and severity indices of 77.72%. The risk factor R6 being ranked 6th of 104 by Managing Directors/Board of Directors; 12th out of 104 by Project Managers; 11th of 104 by Developers and 7th out of 104 by IT support staffs. It has an overall ranking of 11th, with the weighted mean of 3.84 and severity indices of 76.85%.

5.5.4. Development stage risk factors

Table 5.12 : Ranking of impact of cost overrun for Development stage.

			Ranking							
Ref	Mean	Standard deviation	Managing Directors	Project Manager	Developer	IT staff	Coefficient of variation	Severity index	Kendall ranking	Overall Ranking
D1	3.26	1.387	24	28	34	66	42.54	65.12	65.21	31
D2	3.62	1.102	14	14	18	13	30.44	72.47	74.45	14
D3	2.29	0.983	58	64	61	52	42.92	45.74	43.21	61
D4	2.54	1.133	45	51	50	44	44.60	50.86	49.94	51
D5	3.28	1.113	19	20	43	19	33.93	65.68	70.63	26
D6	2.67	0.973	50	46	45	42	36.44	53.4	54.34	46
D7	2.31	0.968	54	60	62	49	41.90	46.11	45.40	57
D8	2.06	0.825	72	74	72	81	40.04	41.3	38.21	73
D9	2.10	1.211	78	71	66	83	57.66	42.04	37.59	72
D10	2.29	0.983	57	63	60	51	42.92	45.74	43.21	60
D11	4.00	1.452	21	5	5	3	36.3	79.94	77.91	4
D12	2.43	0.854	53	53	54	46	35.14	48.58	46.96	53
D13	3.20	1.061	48	21	27	55	33.15	63.95	65.52	36
D14	2.14	1.073	70	56	68	88	50.14	42.72	38.50	69
D15	3.90	1.182	10	11	7	10	30.30	77.96	80.76	8
D16	2.05	0.953	82	73	70	85	46.48	40.99	37.19	75
D17	3.12	1.458	40	37	33	48	46.73	62.47	61.78	41
D18	3.24	1.210	35	32	29	37	37.34	64.75	65.19	32
D19	1.95	0.751	79	83	83	82	38.51	38.95	33.56	82
D20	1.81	0.817	89	91	91	90	45.13	36.23	29.50	90
D21	1.86	0.920	91	89	87	93	49.46	37.16	34.16	89
D22	2.55	1.116	51	50	49	34	43.76	50.99	50.06	50
D23	3.94	0.766	7	8	8	8	19.44	78.7	81.68	7
D24	2.75	1.210	38	44	46	29	44.00	54.94	53.47	45
D25	2.66	0.967	41	47	47	31	36.35	53.27	57.44	47
D26	2.52	1.114	49	52	52	38	44.20	50.49	52.86	52
D27	1.48	0.706	100	100	100	100	47.70	29.69	24.36	100
D28	3.17	1.309	22	42	21	58	41.29	63.4	62.61	39
D29	3.62	1.296	13	15	16	16	35.80	72.35	71.35	15
D30	2.02	1.084	83	75	75	72	53.66	40.49	40.62	79
D31	1.65	0.662	92	95	96	86	40.12	33.02	28.85	95
D32	1.64	0.827	97	96	95	91	50.42	32.72	28.66	96
D33	2.03	1.086	80	76	76	73	53.49	40.56	40.73	76
D34	1.23	0.418	103	103	103	103	33.98	24.51	16.33	103
D35	1.71	0.812	95	93	93	87	47.48	34.2	30.03	93

In the Development stage, 6 of the 35 factors in that stage being ranked in the first 30 highest ranked risk factors. These factors have an overall mean in the range of 3.28 to 4.0. The severity indices are in the range of 65.68% - 79.94%. Factor D11 seemed to be the most important risk factor for this stage and has an overall rank of 4th out of 104 indicators, and the severity indices of 79.94%. The 3 group of practitioners (Project Managers, Developers, IT support staffs) also ranked factor D11 as their top 5 highest ranked risk factor, with a low coefficient of variation of 36.3%. Only Managing Directors/Board of Directors ranked this factor as 21st out of 104.

5.5.5. Implementation stage risk factors

Table 5.13 : Ranking of impact of cost overrun for Implementation stage.

			Ranking							
Ref	Mean	Standard deviation	Managing Directors	Project Manager	Developer	IT staff	Coefficient of variation	Severity Index	Kendall ranking	Overall Ranking
IM1	2.02	1.024	77	80	79	70	50.69	40.31	34.63	78
IM2	2.31	0.968	55	62	64	50	41.90	46.11	45.40	58
IM3	1.94	1.053	75	85	85	84	54.27	38.83	32.65	84
IM4	1.93	0.904	84	86	82	88	46.83	38.52	33.65	86
IM5	1.93	0.737	87	84	80	89	38.18	38.58	33.69	85
IM6	2.43	0.854	56	54	55	47	35.14	48.58	46.98	54
IM7	1.53	0.696	98	99	99	97	45.49	30.68	21.21	99
IM8	3.09	1.221	42	39	36	45	39.51	61.85	63.15	42
IM9	3.15	1.201	32	40	42	27	38.12	62.9	66.30	40
IM10	2.39	1.115	60	57	53	57	46.65	47.84	46.05	55
IM11	2.15	0.944	64	70	71	61	43.90	43.02	38.70	68
IM12	3.58	1.569	43	13	12	12	43.82	71.6	69.22	17
IM13	3.85	1.064	11	10	10	9	27.63	76.98	79.33	10
IM14	3.60	1.371	15	17	15	14	38.08	72.1	72.04	16
IM15	3.26	1.298	28	34	38	26	39.81	65.19	65.55	30
IM16	2.17	0.954	73	67	63	78	43.96	43.4	39.90	67
IM17	2.27	0.977	66	61	57	59	43.03	45.49	43.06	63
IM18	2.19	1.007	74	65	59	79	45.98	43.77	40.92	66
IM19	3.33	1.053	31	27	25	30	31.62	66.67	67.84	22
IM20	3.31	1.114	29	29	28	32	33.65	66.17	68.57	24
IM21	3.30	0.896	52	23	26	22	27.15	65.93	70.95	25
IM22	3.43	0.900	25	24	22	23	26.23	68.58	73.37	21
IM23	3.54	1.255	20	19	14	17	35.45	70.86	75.28	18
IM24	2.90	0.914	36	43	44	33	31.51	58.09	59.75	44

As for the implementation stage, only 9 of the 24 factors in this stage were in the first 30 highest ranked indicators namely; IM12, IM13, IM14, IM15, IM19, IM20, IM21, IM22 and IM23. The overall mean for these factors were in the range of 3.26 to 3.85. The range for the severity indices was 65.19% to 76.98%. All these 9 factors have an overall weighted mean of rank 10th to rank 30th. The factor IM13 was the most important risk factor in this stage with the overall rank 10th out of 104.

5.5.6. Operation maintenance stage risk factors

Table 5.14 : Ranking of impact of cost overrun for Operation & maintenance stage.

			Ranking							
Ref	Mean	Standard deviation	Managing Directors	Project Mgr	Developer	IT staff	Coefficient of variation	Severity index	Kendall ranking	Overall Ranking
OP1	3.20	1.118	37	36	35	35	34.93	64.01	64.11	35
OP2	2.14	0.776	68	69	73	66	36.26	42.78	43.03	70
OP3	1.95	1.022	86	81	78	92	52.41	39.01	38.20	83
OP4	3.28	1.249	23	33	40	40	38.07	65.68	66.85	27
OP5	1.92	0.660	81	88	89	74	34.37	38.33	34.06	87
OP6	2.19	0.635	63	68	69	62	28.99	43.83	43.13	65
OP7	2.19	0.636	67	66	67	63	29.04	43.89	42.34	64
OP8	1.71	0.692	90	94	94	95	40.46	34.32	30.84	94
OP9	1.31	0.465	102	102	102	102	35.49	26.3	19.40	102
OP10	1.44	0.497	101	101	101	101	34.51	28.7	21.77	101
OP11	2.06	1.089	66	77	77	76	52.86	41.11	38.89	74
OP12	1.57	0.720	96	98	98	98	45.85	31.42	26.29	98
OP13	3.18	0.994	39	35	30	43	31.25	63.52	66.27	37
OP14	1.76	1.122	94	92	92	94	63.75	35.35	31.10	92
OP15	1.58	0.723	99	97	97	99	45.75	31.54	25.99	97
OP16	3.32	1.083	34	25	23	36	32.62	66.48	68.67	23
OP17	1.81	0.936	93	90	88	96	51.71	36.3	33.69	91
OP18	3.27	1.085	26	26	31	69	33.18	65.49	66.61	28
OP19	1.23	0.418	104	104	104	104	33.98	24.51	16.33	104

For the Operation and Maintenance stage, only 3 factor (OP4, OP16, OP18) were ranked in the highest 30 ranked risk factor. It has an overall rank of 27th, 23rd and 28th, respectively. The mean range is from 3.27 to 3.32. The severity indices is from 65.49% to 66.48%. The risk factor, OP16 seemed to be the most important factor for this stage. Other risk factors were consistently rated low in the rank by all practitioners.

5.6. Kendal's Concordance analysis

In this research, Kendall's coefficients indicate the level of agreement among the practitioners to the questionnaire on the ranking of risk factors in software development project. If the computed value of significance level is less than 0.05, it indicates that the null hypothesis (there is no agreement between respondents) has to be rejected.

The alternative hypothesis that, there is a significant agreement among the practitioners is acceptable with confidence of ($p > 95\%$). The statistical results of calculated coefficient of variation indicate that there is a variation in practitioner's responses. According to the results shown in Kendal concordance analysis, the data are reliable because the significance value is less than 0.05. Therefore there is a strong agreement between the surveyed practitioners.

Table 5.15 : Kendall's concordance analysis for likelihood occurrence of risks

Stages	Degree of freedom	Chi-square	Kendal's coefficient (W)	Significance
Feasibility study	4	347.780	0.265	0.000
Project planning	13	2066.780	0.491	0.000
Requirement	6	756.379	0.389	0.000
Development	35	5335.278	0.484	0.000
Implementation	23	3011.626	0.404	0.000
Operation & Maintenance	18	2130.661	0.365	0.000
Overall	103	15638.174	0.469	0.000

Table 5.16 : Kendall's concordance analysis for impact of risks on cost overrun

Stages	Degree of freedom	Chi-square	Kendal's coefficient (W)	Significance
Feasibility study	4	371.010	0.286	0.000
Project planning	13	2298.283	0.546	0.000
Requirement	6	1004.629	0.517	0.000
Development	35	4032.560	0.366	0.000
Implementation	23	3123.342	0.419	0.000
Operation & Maintenance	18	2464.576	0.423	0.000
Overall	103	14660.586	0.439	0.000

5.7. Summary

Ranking helps researchers to indicate which risk factors are more important. In this chapter, ranking based on severity index, average weighted mean and standard deviation of each risk factor were used in order to determine the degree of significance on risk factors in the context of software development project. As can be seen from the tables, there were some agreements and disagreements in the rating and ranking of the risk factors among the practitioners. This will be discusses in greater details in the next few chapters.

CHAPTER 6 Software practitioners' perceptions on the likelihood occurrence of risks and their impact on cost overrun of software development project.

6.1. Introduction

While organizations invest substantial time, resources and effort on software development projects, managing and controlling the risks associated with software projects is crucial and critical area of concern (Wen & Sun, 2007; Kumar, 2002; Wallace et al, 2004). Most of the previous studies undertaken explore the relationship of risks factors with the overall project performance and correlate these issues within the perspectives of the project managers, users and organization as a whole. Little has been done, however, to explain the perceptions among different professionals within the software development project team of the likelihood occurrence of these risks and their impact on the cost overrun of the software project in each stages of the lifecycle.

Software projects can often spiral out of control that exceed their original budget, overrun their scheduled due date and not performing in the way expected. The majority of these projects are eventually abandoned or significantly redirected without delivering intended business value. Because of the strategic importance of software projects and the large amount of money and resources involved, it is very crucial to be able to manage the risks of failures or abandonment.

This chapter examines the mindset of professionals in a software development project team as to how they value and perceived each risk factors and its likelihood of occurrence in each stages of the development lifecycle. Due to the large amount of capital and resources involved in a software project, comparisons were also made of the impact of these risks on the cost overrun of a software project. The results of this analysis should be valuable for members of the software development team in understanding the importance of risk occurrence in each stages of the development lifecycle, and crucially, be able to identify and manage the risks that have significant impact on the cost overruns of a software project.

6.2. Findings

6.2.1. Experiences

Table 6.1 and 6.2 below shows the background experience of the respondents :-

Table 6.1 : Experiences of respondents in software development project (Years)

Respondents	Years of experience in software development project						
	< 3	3-6	7-10	11-14	15-18	>18	Number of respondents
Managing Directors / Board of Directors	1	4	9	7	21	4	46
Project Manager	10	26	24	32	31	12	135
Developer/Programmer	12	15	18	33	27	13	118
IT support staff	0	2	4	9	7	3	25
Total	23	47	55	81	86	32	324
%	7.1	14.5	17.0	25	26.5	9.9	

Table 6.2 : Numbers of software development project undertaken by respondents

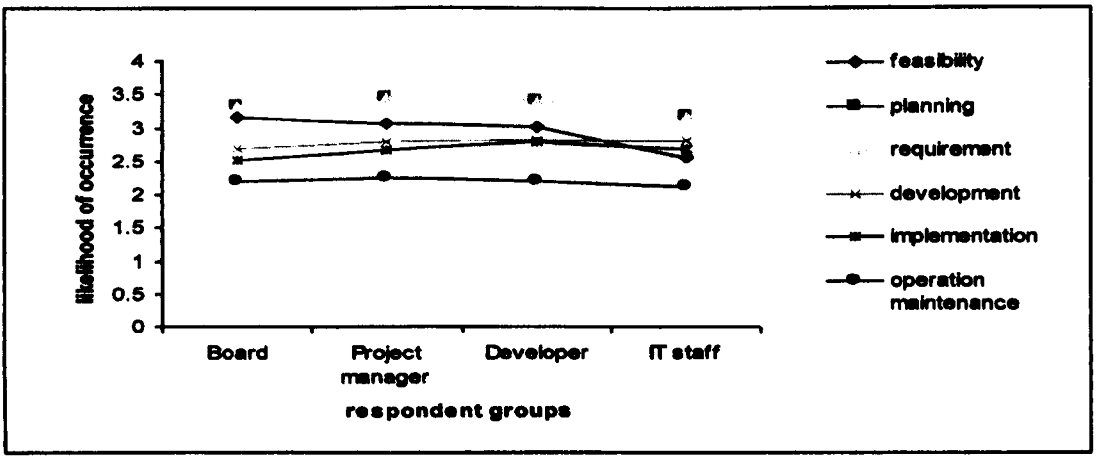
Respondents	Numbers of software development project involved/undertaken						
	< 3	3-6	7-10	11-14	15-18	>18	%
Managing Directors / Board of Directors	1	22	11	3	8	1	14.2
Project Manager	16	43	28	23	16	9	41.7
Developer/Programmer	10	31	31	20	16	10	36.4
IT support staff	0	7	13	2	0	3	7.7
Total	27	103	83	48	40	23	
%	8.3	31.8	25.6	14.8	12.3	7.1	

It can be seen that more than half of the respondents had more than 10 years of experience in software development practice with the overall average of 11.8 years and the standard deviation of 5.29. Furthermore the overall average numbers of software projects that these respondents involved or undertake were 9.2 projects and standard deviation of 5.31. From Table 6.1 and Table 6.2, it can be said that the respondents have good working knowledge and insights of software development projects and processes, in terms of years of experiences and number of projects involved. The wealth of experiences among the respondents was very relevance and significance in justifying the responses that were given in the questionnaires. This may give reasonable support for the arguments in this study.

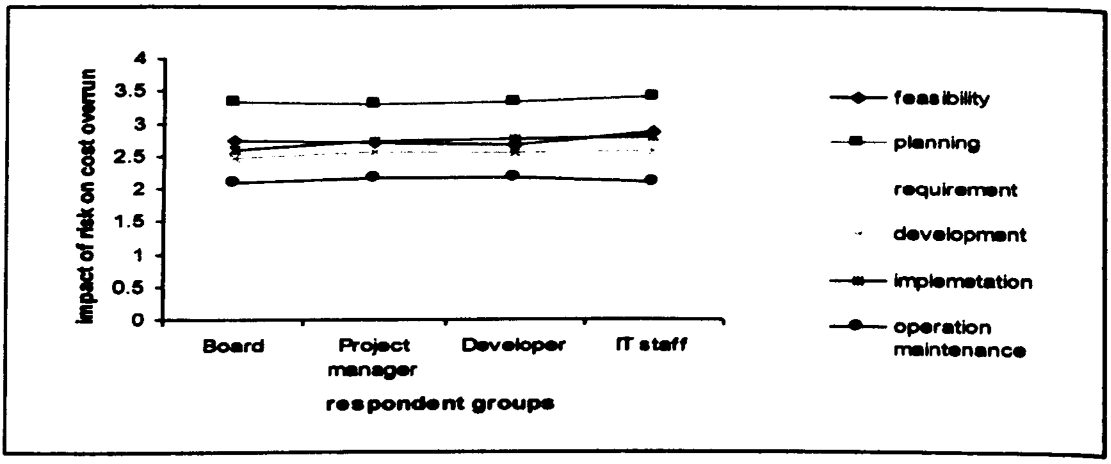
6.2.2. Overall observation

The overall perspectives of the average rating for the stages are shown in Graph 6.1, and Graph 6.2. For the likelihood of occurrence in Graph 6.1, all respondents rated the risks factors in the early part of the development life cycle, higher than the factors towards the end of the development life cycle. That is, to say that they rated the risks factors in the feasibility study, project planning and requirement stages, higher than they rated the risks factors in the development stage, implementation stage and maintenance stage. In general, the ranking for stages was; 1-Project planning (mean average: 3.403); 2-Feasibility study (3.028); 3-Requirement (3.017); 4-Development (2.784); 5-Implementation (2.698); 6-Operation & Maintenance (2.219); respectively.

As for the impact of these risks on the cost overrun of a software project, the average rating is shown in Graph 6.2. The overall rating showed that Project Planning stage (mean average: 3.325) was ranked top, followed by the Requirement stage (3.03) and Implementation stage (2.733). The Feasibility stage (2.712) and the Development stage (2.545) were moderately rated. Where as, the Operation and maintenance stage was rated very low average of 2.164. Based on these results, all 4 groups of respondents agreed that Project Planning stage and the Requirement stage were the two most important stages that have a higher impact on the cost overrun of a software project.



Graph 6.1 : The average rating for Likelihood of occurrence of risk factors



Graph 6.2 : The average rating of risk for Impact of risk factors on cost overrun

6.2.3. Highest 20 ranking

For the purpose of this chapter and more manageable discussions, the risk factors rated by the respondents were ranked and the top 20 ranked risks by each category of practitioners were shown in Table 6.3 and Table 6.4. The full ranked of the risk factors were already highlighted in previous Chapter 5 (Descriptive statistics and ranking).

For the likelihood of occurrence, the risks factors in the project planning and development stages dominate the top end of the ranking. It can be said that these 20 rated risks factors, predominantly were non-technical risks, or some literatures might categorized as organizational issues or project management issues. There were some contrasting trends in the way these practitioners rated these factors, especially on the opinions of the Managing Directors/Board of Directors, the Developers/Programmers and the IT support staff. Although the results from the perspectives of Project Managers tends to agree with the results of some of the previous research and literatures reported, but still there are a few different views.

As for the impact of these risks factors on the cost overrun of a software project, the risk factors such as unclear project scope, unrealistic project schedule, inaccurate estimate of resources dominates the top 3 rated risks with an average rating of 4.24 to 4.33. This showed the important of these risks and the criticality of the Planning stage. The systems requirements factors were also rated high in the rank. The risks factors such as unclear systems requirements, misinterpretations of the requirements and inadequate validation of the requirements, have an average rating of a maximum of 3.99 and a minimum of 3.84. Even though, not as high as the factors in the Planning stage, but to be rated close to 4 is very crucial indeed. Although, the development stage was not rated very high in general, but factors such as inappropriate development methodology used, the inexperienced of the project managers and the lack of control and coordination within the project, still considered as important and contributing factors to the cost overrun of a project.

Table 6.3 : The mean average rating and standard deviation (SD) for Likelihood occurrence of risk factors

Stages	Risk factors	Ref	Managing Directors		Project Manager		Developer		IT staff		Overall	
			Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Feasibility study	Inproper justification of cost benefit analysis and evaluation criteria from feasibility study	F1	4.196	0.98	3.578	0.95	3.517	0.834	3.04	0.79	3.60	0.94
	Too narrow focus on technical IT issues	F2	3.804	0.687	3.096	0.809	3.644	0.822	2.4	0.5	3.34	0.871
	Overlooked the management and business impact issues	F3	3.152	0.868	3.844	1.165	3.356	0.911	2.6	0.5	3.47	1.06
Project planning	Unclear project scope + objectives	P1	4.087	1.071	4.126	1.102	4.195	1.072	4.44	0.87	4.17	1.07
	Undefined project success criteria	P2	3.413	0.748	3.681	0.895	3.771	0.778	3.8	0.764	3.69	0.829
	Inaccurate estimate of resources	P6	4.5	0.658	4.25	1.056	4.602	0.615	4.52	0.77	4.44	0.854
	Unrealistic project schedule	P7	4.37	0.928	4.393	0.774	4.432	0.745	4.48	0.77	4.41	0.784
	Critical and non-critical activities of project not identified	P9	3.848	0.816	3.963	0.98	3.695	1.009	2.48	0.653	3.73	1.019
	Unclear line of decision making authority throughout the project	P11	3.391	0.774	4.585	0.628	3.669	0.878	3.44	0.507	3.99	0.897
	Lack of contingency plan/back up	P12	3.783	0.619	3.622	0.487	3.695	0.745	3.48	0.51	3.66	0.615
Requirement	Unclear and inadequate identification of systems requirements	R1	3.891	0.849	4.052	0.766	4.144	0.719	4.28	0.792	4.08	0.767
	Inadequate validation of the requirements	R6	2.761	0.736	3.978	1.003	3.983	0.773	4.12	0.833	3.82	0.974
Development	Inappropriate development methodology	D2	3.609	0.977	4.23	0.810	4.297	0.788	4.16	0.8	4.16	0.854
	High level of technical complexities	D5	2.652	0.706	3.919	0.906	4.042	0.778	4.00	0.764	3.79	0.944
	Time consuming for testing	D12	3.717	0.807	3.652	0.957	3.729	0.747	3.72	0.792	3.69	0.849
	Resources shifted from project during development	D13	4.217	0.814	3.178	0.953	3.237	1.107	3.24	0.879	3.35	1.047
	Lack of users involvement and commitment	D15	3.957	1.154	3.881	1.146	3.89	1.175	4.24	0.879	3.92	1.139
	Ineffective communication within development team members	D17	4.326	0.871	4.215	0.925	4.28	0.923	4.76	0.523	4.3	0.9
	Ineffective and inexperienced project manager	D23	3.761	0.794	4.096	0.69	4.025	0.698	4.04	0.79	4.02	0.721
	Frequent staff turnover within project team	D24	3.022	1.0	2.985	0.992	3.907	0.654	3.64	0.86	3.38	0.973
	Excessive schedule pressure and overworked	D28	2.891	0.849	4.111	0.852	4.178	0.758	4.36	0.757	3.98	0.924
	Lack of control and coordination within project	D29	3.63	1.062	3.815	1.052	3.797	0.843	3.88	1.236	3.79	0.996
	Lack of regular reviews against goals	D32	4.00	0.789	3.993	0.902	4.161	0.704	4.4	0.645	4.09	0.806
Implementation	Large number of interfaces to other system required	IM9	3.022	1.064	2.904	0.969	3.924	0.730	3.92	0.812	3.37	1.016
	Time constraints of training	IM15	3.674	0.802	3.763	0.975	3.847	0.980	3.84	0.645	3.79	0.841
	Projects affects large number of user departments	IM23	3.522	1.07	3.756	1.168	3.763	0.834	3.56	0.768	3.71	1.015
	Inadequate documentation for implementation	IM24	2.674	1.034	2.785	0.949	3.822	0.823	4.08	0.400	3.25	1.044
Operation Maintenance	Changes in market condition and organisation priorities	OP13	4.065	1.063	3.311	1.068	3.364	0.921	2.64	0.569	3.39	1.036

* Note: The shaded risk factors denotes the 20 highest ranked by each category of practitioner.

Table 6.4 : The mean average rating and standard deviation (SD) for Impact of risk factors on Cost Overrun

Stages	Risk factors	Ref	Managing Directors		Project Manager		Developer		IT staff		Overall	
			Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Feasibility study	Inproper justification of cost benefit analysis and evaluation criteria from feasibility study	F1	3.5	1.130	3.185	1.235	3.153	1.167	3.48	1.358	3.24	1.208
	Overlooked the management and business impact issues	F3	3.304	0.741	3.407	1.128	3.492	0.959	3.44	0.821	3.44	0.995
Project planning	Unclear project scope + objectives	P1	4.17	0.643	4.35	0.480	4.34	0.643	4.44	0.507	4.33	0.572
	Undefined project success criteria	P2	3.76	0.673	3.55	0.798	3.75	0.886	3.8	0.645	3.67	0.809
	Project milestones for stages not well establish	P4	3.48	0.752	2.726	0.981	3.5	1.076	3.6	0.577	3.18	1.035
	Improper change management planning	P5	4.15	0.666	4.16	0.704	3.39	0.774	3.36	0.700	3.82	0.818
	Inaccurate estimate of resources	P6	4.13	0.749	4.03	0.992	4.47	0.770	4.56	0.712	4.24	0.886
	Unrealistic project schedule	P7	4.37	0.771	4.48	0.721	4.13	0.812	4.28	0.737	4.32	0.776
	Critical and non-critical activities of project not identified	P9	3.59	0.717	3.49	1.028	3.53	0.792	3.68	0.748	3.53	0.884
	Lack of contingency plan/back up	P12	4.2	1.108	4.18	0.976	4.22	1.047	4.32	0.988	4.21	1.019
Requirement	Unclear and inadequate identification of systems requirements	R1	3.98	0.577	3.93	0.979	4.01	0.852	4.24	0.723	3.99	0.867
	Misinterpretations of systems requirements	R3	3.89	0.640	3.88	0.890	3.86	0.826	4.00	0.816	3.89	0.827
	Inadequate validation of the requirements	R6	4.02	0.774	3.77	1.593	3.78	1.581	4.2	0.913	3.84	1.456
Development	Inappropriate development methodology	D2	3.67	1.097	3.64	1.231	3.52	0.922	3.92	1.152	3.62	1.102
	High level of technical complexities	D5	3.44	0.860	3.44	1.097	2.975	1.216	3.64	0.757	3.28	1.113
	Failure of user acceptance test	D11	3.391	1.066	4.08	1.511	4.05	1.529	4.36	1.114	4.00	1.452
	Lack of users involvement and commitment	D15	3.85	0.894	3.8	1.292	4.0	1.177	4.04	1.060	3.90	1.182
	Ineffective and inexperienced project manager	D23	4.00	0.471	3.9	0.800	3.9	0.799	4.16	0.850	3.94	0.766
	Lack of control and coordination within project	D29	3.72	1.167	3.57	1.558	3.6	0.988	3.76	1.300	3.62	1.296
Implementation	Failure to manage end-user expectation	IM12	2.587	1.199	3.67	1.647	3.76	1.534	3.96	1.306	3.58	1.569
	User resistance to change	IM13	3.8	0.806	3.82	1.145	3.84	1.101	4.12	0.833	3.85	1.064
	Feedback from users not properly analysed	IM14	3.63	1.466	3.52	1.371	3.64	1.387	3.84	1.143	3.60	1.371
	Projects affects large number of user departments	IM23	3.41	1.257	3.46	1.392	3.65	1.135	3.72	0.980	3.54	1.255

* Note: The shaded risk factors denotes the 20 highest ranked by each category of practitioner.

6.3. Anova analysis.

Based on the calculation of the means, standard deviations and variations, the values of these measurements were quite close together for all 4 groups (Managing Directors/Board of Directors, Project manager, Developers/Programmers, IT staff) of respondents. Since the respondents were asked to give a rating of between the values of 1 to 5, there seems to be not very much significant differences between the groups' responses. An ANOVA analysis was conducted in order to justify the groups' responses statistical differences. Using the SPSS software and with a significance level of 0.05, the hypothesis test was :-

H_0 ($p > 0.05$) : There is no significant differences among the respondents rating for the likelihood of risk factors occurrence.

H_1 ($p < 0.05$) : At least one of the groups rating for the risk factors occurrence significantly different from at least one other groups.

By using the risk factors for likelihood of occurrence in Table 6.3 as an example, the output of the ANOVA analysis of each risk factors is shown in Table 6.5.

The ANOVA in Table 6.5 above shows whether the overall *Fs* values for these risk factors were significant. A statistically significant difference was found among some of the risks factors for the respondents. The 4 group of respondents' responses differ significantly on some of the risk factors such as F1, F2, F3, P6, P9, P11, R6, D2, D5, D13, D17, D24, D28, IM9, IM24 and OP13.

By using the critical values of the *F*- distribution table, the F-values for risk factors that differ significantly was higher than the critical F-values ($F_{critical} = 2.63$ – using the F-Critical value table). In this case, null hypothesis is rejected. This means that at least one of the groups rating for the risk factors occurrence significantly different from at least one other groups.

However, ANOVA analysis only tells whether there is sufficient evidence to state that the rating for risk factors by one group differ significantly from other. It will not tell which specific means are different from which other ones. In order to know this, a follow up test called Post Hoc Multiple Comparison Test. As there were a few post hoc tests that are built into the SPSS, the Tukey test was used for this as the sample size is uneven. For this Tukey test, the risk factors that have the overall F-values are significant are used. But, as a matter of space and content of this chapter, only those factors that show significant difference were shown in Table 6.6.

Table 6.5 : ANOVA analysis for the rating of the likelihood of occurrence risk factors

		Sum of Squares	df	Mean Square	F	Sig.
F1	Between Groups	25.040	3	8.347	10.249	* *0.000*
	Within Groups	260.599	320	0.814		
	Total	285.639	323			
F2	Between Groups	50.934	3	16.978	27.999	* *0.000*
	Within Groups	194.038	320	0.606		
	Total	244.972	323			
F3	Between Groups	44.031	3	14.677	14.736	* *0.000*
	Within Groups	318.719	320	0.996		
	Total	362.750	323			
P1	Between Groups	2.475	3	0.825	0.719	0.541
	Within Groups	367.188	320	1.147		
	Total	369.664	323			
P2	Between Groups	4.611	3	1.537	2.264	0.081
	Within Groups	217.278	320	0.679		
	Total	221.889	323			
P6	Between Groups	8.182	3	2.727	3.837	* *0.010*
	Within Groups	227.457	320	0.711		
	Total	235.639	323			
P7	Between Groups	.297	3	0.099	0.160	0.923
	Within Groups	198.108	320	0.619		
	Total	198.404	323			
P9	Between Groups	47.166	3	15.722	17.469	* *0.000*
	Within Groups	288.007	320	0.900		
	Total	335.173	323			
P11	Between Groups	83.991	3	27.997	50.904	* *0.000*
	Within Groups	175.997	320	0.550		
	Total	259.988	323			
P12	Between Groups	2.101	3	0.700	1.864	0.136
	Within Groups	120.229	320	0.376		
	Total	122.330	323			
R1	Between Groups	3.229	3	1.076	1.845	0.139
	Within Groups	186.684	320	0.583		
	Total	189.914	323			
R6	Between Groups	60.347	3	20.116	26.176	* *0.000*
	Within Groups	245.909	320	0.768		
	Total	306.256	323			
D2	Between Groups	16.838	3	5.613	8.208	* *0.000*
	Within Groups	218.817	320	0.684		
	Total	235.654	323			
D5	Between Groups	70.402	3	23.467	34.554	* *0.000*
	Within Groups	217.327	320	0.679		
	Total	287.728	323			
D12	Between Groups	.425	3	0.142	0.195	* *0.900*
	Within Groups	232.325	320	0.726		
	Total	232.750	323			
D13	Between Groups	40.414	3	13.471	13.752	* *0.000*
	Within Groups	313.475	320	0.980		
	Total	353.889	323			

D15	Between Groups Within Groups Total	2.926 416.145 419.071	3 320 323	0.975 1.300	0.750	0.523
D17	Between Groups Within Groups Total	6.345 255.210 261.556	3 320 323	2.115 0.798	2.652	***0.049**
D23	Between Groups Within Groups Total	3.887 164.001 167.889	3 320 323	1.296 0.513	2.528	0.057
D24	Between Groups Within Groups Total	61.379 244.683 306.062	3 320 323	20.460 0.765	26.757	***0.000**
D28	Between Groups Within Groups Total	65.076 210.813 275.889	3 320 323	21.692 0.659	32.927	***0.000**
D29	Between Groups Within Groups Total	1.459 318.846 320.306	3 320 323	0.486 0.996	0.488	0.691
D32	Between Groups Within Groups Total	4.647 204.933 209.580	3 320 323	1.549 0.640	2.419	0.066
IM9	Between Groups Within Groups Total	78.676 254.880 333.556	3 320 323	26.225 0.796	32.926	***0.000**
IM15	Between Groups Within Groups Total	1.168 227.138 228.306	3 320 323	0.389 0.710	0.548	0.650
IM23	Between Groups Within Groups Total	2.801 329.928 332.728	3 320 323	0.934 1.031	0.906	0.439
IM24	Between Groups Within Groups Total	100.265 251.982 352.247	3 320 323	33.422 0.787	42.443	***0.000**
0P13	Between Groups Within Groups Total	35.947 310.828 346.775	3 320 323	11.982 0.971	12.336	***0.000**

Table 6.6 : Post Hoc test for the rating of the likelihood of occurrence risk factors

F1

Roles	N	Subset for alpha = 0.05		
		1	2	3
IT staff	25	3.04		
developer	118		3.52	
project manager	135		3.58	
board of directors	46			4.20

F2

Roles	N	Subset for alpha = 0.05		
		1	2	3
IT staff	25	2.40		
project manager	135		3.10	
developer	118			3.64
board of directors	46			3.80

F3

Roles	N	Subset for alpha = 0.05		
		1	2	3
IT staff	25	2.60		
board of directors	46		3.15	
developer	118		3.36	
project manager	135			3.84

P9

Roles	N	Subset for alpha = 0.05	
		1	2
IT staff	25	2.48	
developer	118		3.69
board of directors	46		3.85
project manager	135		3.96

P11

Roles	N	Subset for alpha = 0.05	
		1	2
board of directors	46	3.39	
IT staff	25	3.44	
developer	118	3.67	
project manager	135		4.59

P12

Roles	N	Subset for alpha = 0.05	
		1	2
IT staff	25	3.48	
project manager	135	3.62	3.62
developer	118	3.69	3.69
board of directors	46		3.80

R1

Roles	N	Subset for alpha = 0.05	
		1	2
board of directors	46	3.89	
project manager	135	4.05	4.05
developer	118	4.14	4.14
IT staff	25		4.28

R6

Roles	N	Subset for alpha = 0.05	
		1	2
board of directors	46	2.76	
project manager	135		3.96
developer	118		3.98
IT staff	25		4.12

D2

Roles	N	Subset for alpha = 0.05	
		1	2
board of directors	46	3.61	
IT staff	25		4.16
project manager	135		4.23
developer	118		4.30

D6

Roles	N	Subset for alpha = 0.05	
		1	2
board of directors	46	2.65	
project manager	135		3.92
IT staff	25		4.00
developer	118		4.04

- Groups listed in the same subset are not significantly different.
- But groups in different subset are significantly different.
- For example, the ratings for F1 are not significantly different between developer and project manager, but significantly different between IT staff and developer, or significantly different between IT staff and Board of directors.

- The mean difference is significant at the 0.05 level.

Table 6.6: Post Hoc test for the rating of the likelihood of occurrence risk factors (continued)

D13

Roles	N	Subset for alpha = 0.05	
		1	2
project manager	135	3.18	
developer	118	3.24	
IT staff	25	3.24	
board of directors	46		4.22

D17

Roles	N	Subset for alpha = 0.05	
		1	2
project manager	135	4.21	
developer	118	4.28	
board of directors	46	4.33	
IT staff	25		4.76

D24

Roles	N	Subset for alpha = 0.05	
		1	2
project manager	135	2.99	
board of directors	46	3.02	
IT staff	25		3.64
developer	118		3.91

D28

Roles	N	Subset for alpha = 0.05	
		1	2
board of directors	46	2.89	
project manager	135		4.11
developer	118		4.18
IT staff	25		4.36

D32

Roles	N	Subset for alpha = 0.05	
		1	2
project manager	135	3.99	
board of directors	46	4.00	
developer	118	4.16	4.16
IT staff	25		4.40

IM9

Roles	N	Subset for alpha = 0.05	
		1	2
project manager	135	2.90	
board of directors	46	3.02	
IT staff	25		3.92
developer	118		3.92

IM24

Roles	N	Subset for alpha = 0.05	
		1	2
board of directors	46	2.67	
project manager	135	2.79	
developer	118		3.82
IT staff	25		4.08

OP13

Roles	N	Subset for alpha = 0.05		
		1	2	3
IT staff	25	2.64		
project manager	135		3.31	
developer	118		3.36	
board of directors	46			4.07

- **Groups listed in the same subset are not significantly different.**
- **But groups in different subset are significantly different.**
- **For example, the ratings for D24 are not significantly different between board of directors and project manager, but significantly different between IT staff and project manager.**

- **The mean difference is significant at the 0.05 level.**

From Table 6.6, with the significant values of less than 0.05 ($p < 0.05$), shows that there is significant difference between the respondents' responses. For example, the ratings for F1 are not significantly different between developer and project manager, but significantly different between IT staff and developer, or significantly different between IT staff and Board of directors. The ratings for D24 are not significantly different between board of directors and project manager, but significantly different between IT staff and project manager.

Using the same concept and application as for the likelihood occurrence in previous section, SPSS software and ANOVA analysis will also be conducted on the risk factors impact on the cost overrun. With the significance level of 0.05, the hypothesis test was :-

H_0 ($p > 0.05$) : There is no significant differences among the respondents rating for the impact of risk factors on the cost overrun.

H_1 ($p < 0.05$) : At least one of the groups rating for the impact of the risk factors on the cost overrun significantly different from at least one other groups.

Using the risk factors for the impact on the cost overrun in Table 6.4, the output of the ANOVA analysis of each risk factors is shown in Table 6.7.

Table 6.7 : ANOVA analysis for the risk factors impact on the cost overrun

		Sum of Squares	df	Mean Square	F	Sig.
F1	Between Groups	5.858	3	1.953	1.343	0.260
	Within Groups	465.365	320	1.454		
	Total	471.222	323			
F2	Between Groups	1.319	3	0.440	0.583	0.627
	Within Groups	241.431	320	0.754		
	Total	242.750	323			
F3	Between Groups	.677	3	0.226	0.227	0.878
	Within Groups	318.962	320	0.997		
	Total	319.639	323			
P1	Between Groups	1.521	3	0.507	1.558	0.200
	Within Groups	104.143	320	0.325		
	Total	105.664	323			
P2	Between Groups	3.642	3	1.214	1.870	0.134
	Within Groups	207.679	320	0.649		
	Total	211.321	323			
P4	Between Groups	48.419	3	16.140	17.341	* *0.000*
	Within Groups	297.838	320	0.931		
	Total	346.256	323			
P5	Between Groups	48.079	3	16.026	30.494	* *0.000*
	Within Groups	168.177	320	0.526		
	Total	216.256	323			
P6	Between Groups	15.114	3	5.038	6.756	* *0.000*
	Within Groups	238.623	320	0.746		
	Total	253.738	323			
P7	Between Groups	8.063	3	2.688	4.610	* *0.004*
	Within Groups	186.554	320	0.583		
	Total	194.617	323			
P9	Between Groups	.937	3	0.312	0.397	0.755
	Within Groups	251.690	320	0.787		
	Total	252.627	323			

P12	Between Groups	.461	3	0.154	0.147	0.932
	Within Groups	334.684	320	1.046		
	Total	335.145	323			
R1	Between Groups	2.042	3	0.681	0.904	0.439
	Within Groups	240.930	320	0.753		
	Total	242.972	323			
R3	Between Groups	.384	3	0.128	0.186	0.906
	Within Groups	220.391	320	0.689		
	Total	220.775	323			
R6	Between Groups	5.841	3	1.947	0.917	0.433
	Within Groups	679.131	320	2.122		
	Total	684.972	323			
D2	Between Groups	3.714	3	1.238	1.020	0.384
	Within Groups	388.348	320	1.214		
	Total	392.062	323			
D5	Between Groups	18.674	3	6.225	5.225	* *0.002*
	Within Groups	381.203	320	1.191		
	Total	399.877	323			
D11	Between Groups	20.286	3	6.762	3.275	* *0.021*
	Within Groups	660.711	320	2.065		
	Total	680.997	323			
D15	Between Groups	3.144	3	1.048	0.748	0.524
	Within Groups	448.495	320	1.402		
	Total	451.639	323			
D23	Between Groups	1.751	3	0.584	0.994	0.396
	Within Groups	187.888	320	0.587		
	Total	189.639	323			
D29	Between Groups	1.296	3	0.432	0.255	0.857
	Within Groups	541.247	320	1.691		
	Total	542.543	323			
IM12	Between Groups	50.221	3	16.740	7.193	* *0.000*
	Within Groups	744.693	320	2.327		
	Total	794.914	323			
IM13	Between Groups	2.036	3	0.679	0.597	0.617
	Within Groups	363.553	320	1.136		
	Total	365.590	323			
IM14	Between Groups	2.600	3	0.867	0.459	0.711
	Within Groups	604.832	320	1.890		
	Total	607.432	323			
IM23	Between Groups	3.923	3	1.308	0.829	0.478
	Within Groups	504.472	320	1.576		
	Total	508.395	323			

The ANOVA in Table 6.7 above shows the overall *Fs* values for these risk factors were significant. A statistically significant difference was found among some of the risks factors for the respondents. The 4 group of respondents' responses differ significantly on some of the risk factors such as P4, P5, P6, P7, D5, D11 and IM12. In this case, null hypothesis should be rejected. This means that at least one of the groups rating for the impact of the risk factors on the cost overrun significantly different from at least one other groups. A follow up Post Hoc Multiple Comparison Test was performed to see the significance difference among the groups of practitioners. The Tukey test was used for this as the sample size is uneven. For this Tukey test, the risk factors that have the overall F-values are significant are used. As shown in Table 6.8, the ratings for P4 are not significantly different between board of directors, developers and IT staff, but significantly different between project manager and other practitioners. The rating for P6 is significantly different between project manager and developer, but not significantly different between developer and IT staff.

Table 6.8 : Post Hoc test for the risk factors impact on cost overrun

P4			
Roles	N	Subset for alpha = 0.05	
		1	2
Project manager	135	2.73	
Board of Directors	46		3.48
Developer	118		3.50
IT staff	25		3.60

P5			
Roles	N	Subset for alpha = 0.05	
		1	2
IT staff	25	3.36	
Developer	118	3.39	
Board of Directors	46		4.15
Project manager	135		4.16

P6				
Roles	N	Subset for alpha = 0.05		
		1	2	3
Project manager	135	4.03		
Board of Directors	46	4.13	4.13	
Developer	118		4.47	4.47
IT staff	25			4.56

D6			
Roles	N	Subset for alpha = 0.05	
		1	2
Developer	118	2.97	
Board of Directors	46	3.43	3.43
Project manager	135	3.44	3.44
IT staff	25		3.64

D11			
Roles	N	Subset for alpha = 0.05	
		1	2
Board of Directors	46	3.41	
Developer	118	4.05	4.05
Project manager	135	4.08	4.08
IT staff	25		4.36

IM12			
Roles	N	Subset for alpha = 0.05	
		1	2
Board of Directors	46	2.63	
Project manager	135		3.67
Developer	118		3.76
IT staff	25		3.96

- Groups listed in the same subset are not significantly different.
- But groups in different subset are significantly different.
- For example, the ratings for P4 are not significantly different between board of directors, developers and IT staff, but significantly different between project manager and other practitioners.
- The mean difference is significant at the 0.05 level.

Considering the ANOVA analysis and Post Hoc comparison test for both the likelihood of occurrence of risk, and the impact of risk on the cost overrun, it can be said that the difference among the means are significant. As the *F*-values were higher than the $F_{critical}$, and level of significance 0.05 ($p<0.05$), means that there is less than 5 in 100 chance that the difference between the practitioners came about by chance. It can be accepted that there is a genuinely significant overall difference among the practitioners in their rating of the risk factors, in terms of likelihood of occurrence and impact on cost overrun.

6.4. Discussion

6.4.1. Common agreements

From Table 6.3 and Table 6.4, there were common agreements among the practitioners with factors such as unclear project scope (P1), inaccurate estimate of resources (P6), unrealistic project schedule (P7) and unclear identification of systems requirements (R1). These 4 factors interchange the top 10 ranking among the practitioners. For the likelihood of occurrence, these 4 factors were rated with a minimum average rating of 4.05 and a maximum average rating of 4.62, which is highly likely to occur. Whereas for the rating of the impact of these risk factors on cost overrun, the minimum average rating was 4.01 and a maximum average rating of 4.56, which shows a high impact on cost overrun.

Although, the definition of resources are quite broad and subjectives, but resources such as financial resources or human resources can have ripple effect on the completion of the project, its scheduling, the quality and the workmanship of the output produced. The inaccurate estimates of resources and unrealistic scheduling might also resulted from the decision made by the top management as to how much they willing to spend on the project and when do they want it to be up and running.

Research studies done by Verner et al (2007) explore the direct effects of the cost and schedule estimation on the success or failure of a software development project. Verner et al (2007) also stressed that not only need to use good estimation methods but also require good requirements, coupled with knowledge of the organization's performance, working practices and software experiences. This suggests that cost estimation in isolation, is not enough. Charette (2005) also asserted that cost and schedule estimation are poorly developed and reflect an unstated assumption that all will go well, suggesting a lack of risk assessment.

A study did by Schmidt et al (2001), highlighted the top management involvement and commitment being the important risks factors rated by the experience project managers from HongKong, USA and Finland. By getting involved and committed to the project from the beginning right to the end, would probably resulted in better estimates of the resources and timeframe of the project. Ropponen (2000) and Ming Han (2007) also stressed the important elements of resources, cost and scheduling, have a significant impact on the performance of a software development project. The study did by Ropponen (2000) discussed the fact that scheduling is a very important risk components that have significant impact on the performance of the software development project. The components highlighted were also includes changes in the schedule, unrealistic schedule and problems in the schedule. The study shows that by applying risk management method, the scheduling risk can be managed significantly better. It also shows the performance of the risk management method seems to improve with general project experience.

Given the pressure under which many project managers must function, better knowledge of the impact of both direct and indirect estimation factors will help them focus effort on project areas that are likely to cause software project failures. Moreover, political agenda in many organizations may also be stacked against the project managers, making it difficult to convince the senior management to accept realistic estimates.

In practical, it is probably almost impossible to pin down the exact scope and requirements at the outset of a project. The final or absolute version of the systems requirements would probably become clearer as the project progresses. It might make sense, that projects involving unstable scope or uncertain systems requirements, generally did not perform well and exceeds budgets. As pointed out by Wallace & Keil (2004), if scope and requirements were identified, execution problems are less important, though they still affect the process and the likelihood of project being completed within budget. Proccacino et al (2005) stated that the project managers should consider ensuring that project requirements are accepted by the team as being realistic and achievable, given the available time, resources and technology. Project managers must be willing to draw a line between desirable and absolute necessary functionality. It might be helpful in establishing scopes and requirements that were not included for the project.

6.4.2. Project Manager's perspectives

As for Project Managers, most of them agreed that, unclear line of decision making authority throughout the project (factor P11) is very highly likely to occur during the software development project. The Project Managers rated this factor with a high average of 4.59. This risk factor may resulted from the lack of management support and commitment. As better management support and commitment could mean better hierarchy of decision making process. The fact that PM is normally the first person in line with the management, probably shows that, the main issues of risks were actually from the top management rather the development team or the development process itself, at least from the perspectives of the PM.

In any project, decision making or changes coming from the top management do tends to be inconsistent as from who it came from, the timing of the decisions, the dateline for the project to be implemented and who has the absolute right to influence any decision in any particular project. This might become worse when there are changes in the management hierarchy, internal politics and even changes in the organization priorities. This result does agree with some of the previous research which concludes that top management involvement and commitment are very important factors in successful completion of a software development project (Jiang & Klein, 2000; Na et al, 2007; Keil et al, 2002; Wen & Sun Jen Huang, 2007).

Wallace & Keil (2004) also argued that top management involvement and commitment was a very important risk factor, but it's beyond the control of the project management development team. This makes it rather more significant for the development team to be clear of who is the absolute authorization hierarchy for decision makes during the project. However, this risk factor was not ranked as high when it comes to its impact on the cost overrun of the software project. Project Managers rated this factor an average of 3.13, which put the risk factor ranked 41st of all 104 risks factors. Speculatively, this can also relate with the response by the Managing Directors/Board of Directors, where they rated the risk of the resources being shifted due to organizational priorities (D13) and changing market conditions (OP13) were highly likely to occur. The risk of the resources being shifted due to organizational priorities (average 4.22) and market conditions (average 4.07), were ranked 4th and 7th respectively in the likelihood of occurrence rating. This shows that the top management themselves do realize the potential of the decision made by the top management being change during the project.

Although Proccacino et al (2002) concluded that it is not necessary for a project success for the Project Manager to be given full authority to manage the project, but the decision making authority must be clearly determined in order to avoid overlapping, mismanagement and conflicting outcomes. Project Managers probably concentrating on the factors that affect their function as project managers, and he/she certainly viewed the decision making authority is crucial for successful project.

6.4.3. Areas of disagreement

As can be seen in Table 6.3, there are a few factors that stand out in a way the Managing Directors/Board of Directors perceived the likelihood occurrence of risks, compared to other practitioners. There were 4 factors that the Managing Directors rated quite high, which none of the other practitioners consider rating them in the top 20, and these 4 risks factors did not even ranked in the top 20 of the overall rating for the likelihood of occurrence :-

i. Resources shifted from project due to organizational priorities. (D13)
ii. Inproper justification of cost benefits analysis. (F1)
iii. Changes in market condition and organizational priorities. (OP13)
iv. Too narrow focus on the technical IT issues. (F2)

This shows that the top management views factors that have a direct bearing on the way the business going to move forward as very important issues. The business responses to the way the market changes to the economics of the demand and supply, the manner in which resources being allocated and utilized, and their competitors' strategies, were very crucial in maintaining the competitive businesses survival. Bear in mind that, at the end of the day, businesses were still about profit making and having the competitive advantage over rivals in the long run.

As for PM, Developers and even IT support staff, the focus of the rankings were more on the processes and product related issues of the development processes. Issues such as experiences of the project managers (D23), the development methodology (D2) used for the project and excessive schedule pressure and overworking (D28) were deemed very important to these practitioners. Keil (2002) viewed the development methodology used as very important as this risk leads to an end-product that may not met the users' requirements. It can also lead to a finished product that is inadequate or lack of quality.

These 3 practitioners were more focus on the main task in hand and not worry too much on the business related issues. Another issue that probably worth mentioning is the factor that was being rated quite high by the Developer and IT support staff, that is, the risk of frequent staff turnover within the project team (D24). This factor was not ranked in the top 20 by the Board and PM in the likelihood of occurrence. As can be expected from any business, staff turnover or staff changes can happen anytime during the course of the project. But from the perspectives of the developers and IT staff, it is not an ideal situation for the development of the project. Of course, there were a lot of reasons as to why the staff left. But when new staff came in the middle of the project, it is rather difficult for the new staff to get the momentum going for the project, as he/she needs the familiarization process and probably some training involved. This might delay the project and put unnecessary pressure or overworked on the staff in meeting the dateline for the project.

As for impact of the risks factors on the cost overrun of a software project in Table 6.4, there were also a few factors that the Managing Directors/Board of Directors perceived differently compared to the other practitioners. The Managing Directors still rated the risk factor, improper justification of the cost benefit analysis (F1), quite high as it did for the likelihood of occurrence. Another factor that was rated very high by the Managing Directors was the factor, improper change management planning (P5). In fact, only the Managing Directors and Project Managers rated this factor very high, but not the Developer and the IT staff. This might be because change management is part of the management and project manager's responsibilities and not the responsibilities of the developers and IT staffs.

Another significant different for risk factors impact on cost overrun in Table 6.4, is that, apart from the Board of Directors/Managing Directors, the other 3 practitioners (Project Managers, Developers, IT staffs) rated two user related issues very high in the rank. The two user related risks factors, failure of user acceptance test (D11) and failure to manage end-user expectation (IM12), were rated a minimum of 3.67 and a maximum rating of 4.36, by the 3 practitioners (Project Managers, Developers, IT staffs). The Managing Directors probably did not see this as very important factors, as they did not involve directly with the users. In comparisons with the other 3 practitioners (Project Managers, Developers, IT staffs), dealings with users and managing their expectations of the end product were very crucial indeed.

6.4.4. Non-technical factors.

From the likelihood occurrence of Table 6.3, it can be seen that the risks factors were predominantly non-technical related issues. Issues of success criteria, users involvement, communications and project management issues dominates the list as being discussed by most of the literatures (Wen & Sun,2007; Jiang & Klein,2000; Wallace & Keil, 2004; Proccaccino et al, 2005).Only 3 risks factors, considered as technical related risks, commonly agreed by all practitioners to be rank in the top 20 of the likelihood of occurrence :

- Unclear and inadequate identification of systems requirements (R1).
- Time consuming for testing. (D12)
- Inappropriate development methodology used. (D2)

In fact, if the list goes beyond this top 20 and considering the highest 50 risks factors, only 24% of the top 50 factors was technical related factors, while others were non-technical issues. Furthermore, 31% of the non-technical factors in the top 50 factors has an average rating of 4 and above (4-highly likely to occur; 5-very highly likely to occur). In contrast, only 16% of the technical related factors in the top 50 have that kind of average rating. In addition, out of the whole 104 factors, 68% of the technical related factors have an average rating of 3 and below, and 47% of the non-technical factors has an average rating as low as that. This shows that these practitioners viewed that risk factors that are not directly related to technical issues should be paid more attention.

As for the impact of these risks factors on the cost overrun of a software project, as shown in Table 6.4, the highest 20 ranked factors, was still dominated by the non-technical related factors. Only 5 technical related risks factors were ranked in the top 20 :-

- Unclear and inadequate identification of systems requirements. (R1)
- Misinterpretations of systems requirements. (R3)
- Inadequate validation of systems requirements. (R6)
- Inappropriate development methodology used. (D2)
- High level of technical complexities. (D5)

From the highest 50 ranked factors on the impact of the factors on cost overrun, 78% of the factors were non-technical issues and 22% was technical related factors. In fact, 18% of the non-technical factors in the highest 50 ranked risks factors was rated 4 and above, and only 10% on the technical factors in the highest 50 ranked risks factors was rated 4 and above. Consequently, for the whole 104 factors, 35% was technical related factors, and 65% was non-technical related factors. In addition, 75% of the technical issues for all 104 factors, have an average rating 3 and below, whereas, only 48% of the non-technical factors have that kind of low average rating.

Although, there is not much empirical evidence to support any definitive and widely accepted definition of non-technical issues, some researchers may have classified as human issues, organizational issues, managerial issues, behavioural issues, business issues or even people issues. Historically, software projects have been preoccupied with technical issues at the expense of the organizational issues. However, there is evidence suggesting that the treatment of organisational issues is perceived as more important than technical issues in determining the successful outcome of software development projects (Ewusi-Mensah et al,1994; Hornby et al,1992).

Typically, organisational issues have been defined by providing examples of the 'non-technical' aspects of system development, which might have an impact on the ultimate success or failure of a project (Clegg et al,1989). In the context of IT development, Doherty and King (1998) suggested a more generic definition of organizational issue, as an interface between the technical system and the characteristics, requirements of organization or its individual employees, which can lead to operational problems within the organization.

While there might be variations in the ways, organization, management or IT people perceived or translate non-technical issues, but perhaps the most important findings would be that the implications of the non-technical issues and the ways the issues were treated need to be carefully considered.

6.4.5. The contrasting ranking

Despite the commonly agreed and disagreement of risks factors mentioned earlier, the survey also resulted in a few risks factors consistently being ranked in a high ranked order such as ineffective communication within development team (D17), inappropriate development methodology used (D2), inexperienced project manager (D23), lack of control (D29) and high level of technical complexities (D5). However, there were a few factors that had rather different ranking for likelihood of occurrence and a very much contrasting rating for the impact of the risks factors on the cost overrun of a software project. Figure 6.1 below shows the overall ranking of risk factors in terms of likelihood of occurrence and impact on the cost overrun.

As shown in Figure 6.1, there were a few risks factors which being rated high in likelihood occurrence but low in impact on cost overrun, and vice versa. Factors such as ineffective communications, inappropriate development methodology used, inexperienced project managers, high level of technical complexities, time consuming for testing, lack of regular reviews against goals, and a few user related risks factors.

Ineffective communications within development team members (D17) was rated an overall average in the top 3 for likelihood of occurrence but was ranked 41st for the impact on cost overrun of software project. It could be said that, ineffective communications was rather common issues that was bound to happen. It is a certainly being accepted as a common risk as communication issues among human being was quite complex and unpredictable. Jiang & Klein (2000) stated that poor communications among development team members does not allow for the coordination necessary to conduct the individual tasks required to complete the project in an orderly fashion. Much of the time might be spent on duplication of efforts and progress will be towards individual's goal rather than the project goal.

The research carried out by Proccaccino et al (2005) also highlighted the importance of actively nurturing effective communication that improves interpersonal relationships of their team members. As pointed out by Linberg (1999) and Glass (1999), particular attention should also be given to internal intrinsic items, as they relate to motivation and productivity. As well as project management expertise, project leaders should also have communication and people skills in managing the software development project.

As for the other factors such as inappropriate development methodology used (D2), inexperienced project managers (D23), high level of technical complexities (D5), time consuming for testing (D12) and lack of regular reviews (D32), these factors were rated quite high when it comes to the likelihood of occurrence, but were rated rather low in terms of their impact on the cost overrun of a project. This could indicate that the project managers or the development team expected these risks factors going to occur. Eventhough, these risks have a greater chance of occurrence, but the Project managers and development team would probably felt that they can control and understand these issues, and as a result, they thought that these risks factors only have minimal impact on the cost overrun of the project. They might already have something in the project plan to counteract these issues in the first place.

Significant contrasting views were the user related risks factors. Factors such as failure of user acceptance test (D11), users resistance to change (IM13) and failure to manage the users expectations (IM12). These factors were rated quite low in the likelihood of occurrence, but were rated rather high in the impact of these risks on the cost overrun of a software project. Although the practitioners thought these factors were very rare to happen, but they could have a drastic impact on the cost overrun, once they occur.

Jiang et al (2002) and Barki et al (1993), suggested user related risks as the extent to which prospective software users participate in software project development, their readiness to accept the proposed software system, their attitude toward the software and their experience in software project development. These issues make it difficult to understand and to predict the users' expectations and requirements, and thus the completion of the final project within the timeframe and the budget allocated. Project management literature also suggested the importance of the communications between the development team and the users in defining the project scope and controlling the project changes. Lapses in these tasks could lead to increased uncertainty throughout the development cycle and could contribute to project overruns. The project managers and the development team need to build, create and maintain good relationship and trust with the users, to avoid being caught in a situation where supports and commitments for the project suddenly evaporates.

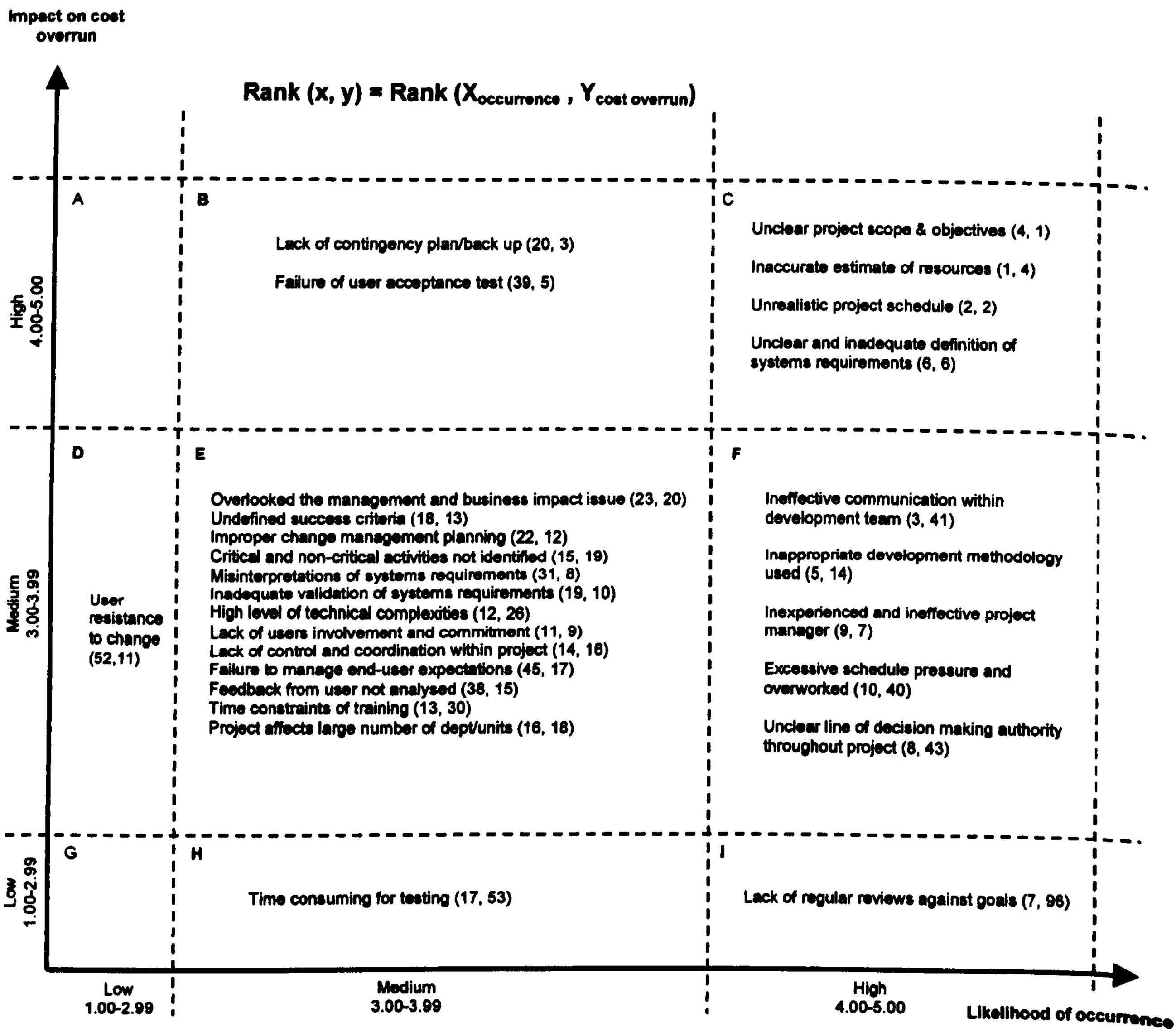


Figure 6.1 : The overall rank of risk factors

Note : Figure above only shows the combination of the top 20 ranked risk factor for the likelihood occurrence and the top 20 of the risk impact on cost overrun.

6.5. Summary

From the study, it is clear that practitioners viewed the risks factors in the project planning phases, requirement phases and development phases are very important stages in the software development project as most risks factors in these 3 stages were ranked high in ranking. The factors such as unclear project scope, estimate of resources, project scheduling and systems requirements were deemed very important issues in dealing with risks in a software development project. This is consistent with the perspectives of success criteria of a project as meeting the budget or resources and timely completion.

Despite a few contrasting views of the ranking of the risks factors for likelihood occurrence and their impact on the cost overrun of a software project, particular attention should also be given to the non-technical related risks factors and the user related factors. These factors should be taken lightly as even they are very unlikely to happen, but when it does happen, they can have a very significant ripple effect on other tasks and processes.

Eventhough, there are a range of issues interact together to have an impact on the success or failures of software project, it is also fair to say, the importance and significance of each risk factors may also be different and dependable on the type of software project. However, the software development project should not be seen as only technology driven, but also as a process driven, and process driven involved a lot of factors from the feasibility study phase of the lifecycle right to the end of the lifecycle of software development.

The differences of risks perceptions based on the role taken within the development team were very important for the coordination among the various groups. This may indicate the need for improved communications in order to develop a shared understanding of project risk. Without a shared understanding of risk, it is unlikely that the Project Manager and their development team will be able to work together effectively, and there may be an increased potential for conflicts to arise.

CHAPTER 7 FACTOR ANALYSIS OF RISK FACTORS IN SOFTWARE DEVELOPMENT PROJECT

7.1. Introduction

Failures of software projects are widely documented in academic research literatures. A lot of the problems relate essentially to complexity, size, requirements, resources and also project management issues. Software development projects are often difficult to estimate and manage and some troubled projects are cancelled or reduced in scope because of overruns in cost and time, loss of motivation and burnout, unpaid overtime or even failure to produce anticipated benefits. (Kumar,2002; Procaccino et al,2002).

Many studies have proven that a proper management of software risks affects the success or failure of a project. (Wen & Sun,2007; Jiang & Klein,2000; Wallace & Keil, 2004). Identifying the software risks that negatively affect the project performance should be well controlled in order to improve the project performance. Failure to understand, identify and manage these risks is often cited as a contributing factor in software project failures.

The purpose of this chapter is to undertake factor analysis and data reduction process from a result of a survey. Based on the factors relationship and correlations, the outcome of the data reduction is presented in a few components that consist of the most important risk factors of the original large group risk factors. At the end of the day, a more clearly and manageable understanding of new cluster or most important list of risk factors, and their implications will be instrumental in the influence of the risk factors on the software development project.

7.2. Factor analysis

Factor analysis is often used in data reduction to identify a small number of factors that explain most of the variance observed in a much larger number of variables (Morgan et al, 2004; Punch, 2005). Having many variables often makes it difficult to understand the data. Factor analysis technique can reduce the number of variables, but without losing too much of the information the original variables provide (Field, 2005; Punch, 2005).

Multicollinearity can be a problem in multiple regressions with a lot of variables, and factor analysis can be used to solve this problem by combining variables that are collinear (Field, 2005). This data reduction is achieved by looking for variables that correlate highly with a group of other variables, but do not correlate with variables outside of that group. This technique is looking at which variables seem to cluster together in a meaningful way. When 2 or more variables are correlated, an existence of a common factor can be proposed, which these variables share to some extent, and which therefore explains the correlation between them.

Factor analysis can be used either in hypothesis testing or in searching for constructs within a group of variables for more easily understood framework. The process begins by finding a linear combination of variables that accounts for as much variation in the original variables. It then finds another component that accounts for as much of the remaining variation as possible and it is uncorrelated with the previous component. The process cycles and 'rotation' (as termed by SPSS) continues in this way until there are as many components as original variables. Usually, a few components will account for most of the variation, and these components can be used to replace the original variables. (Punch, 2006; Morgan, et al 2004; Field, 2005). By reducing a data set from a group of interrelated variables into a smaller set of factors, factor analysis achieves parsimony by explaining the maximum amount of common variance in a correlation matrix using the smallest number of explanatory concepts (Field, 2005).

The degree of significance of each risk factor in software development project varies according to its influence, occurrence and impact on the software project. It can be said that some risk factors can be influential in comparisons with others. A number of risk factors with the highest degree of significance might be considered as representative of the whole set of data. Consequently, the most important risk factors are extracted and are treated as representative of the whole set of risk factors.

7.3. Factor analysis process

For undertaking these analyses of the data, Statistical Package for the Social Science (SPSS) and Microsoft Excel were used. In SPSS, the principal components method is used to extract the latent components and variables (Morgan et al, 2004; Field, 2005). Components are a set of matrixes that present the correlations between different variables.

The first stage of the factor analysis is to determine the strength of the relationship among the variables. A matrix of correlation coefficients should be extracted, and then components, carrying Eigen value of 1 and more should be extracted from matrix of correlation coefficient (the most common extraction method is based on principal component analysis). In the next phase, a rotated component matrix should be generated in order to determine which risk factors have more effective influence in each component.

The existence of 104 risk factors in this survey makes it difficult to handle the analysis, therefore factor analysis and data reduction are considered as an important process to decrease the number of risk factors in order to handle the task more efficient. This aim was achieved through the application of SPSS software and as a result the redundant data is removed from the list of questions in order to obtain a manageable subset of the risk factors that represent the majority of the risk factors. The process of the analysis is shown in Figure 7.1. The figure shows that through the use of data reduction in SPSS, 104 risk factors in 6 stages of lifecycle are analysed to a few components. The correlations and interactions of risk factors with each other are computed. The SPSS software helps to analyse the factors, and categorises the factors according to their relationship and correlations to each other. The process, findings and discussions of the data analysis are presented in the following sections.

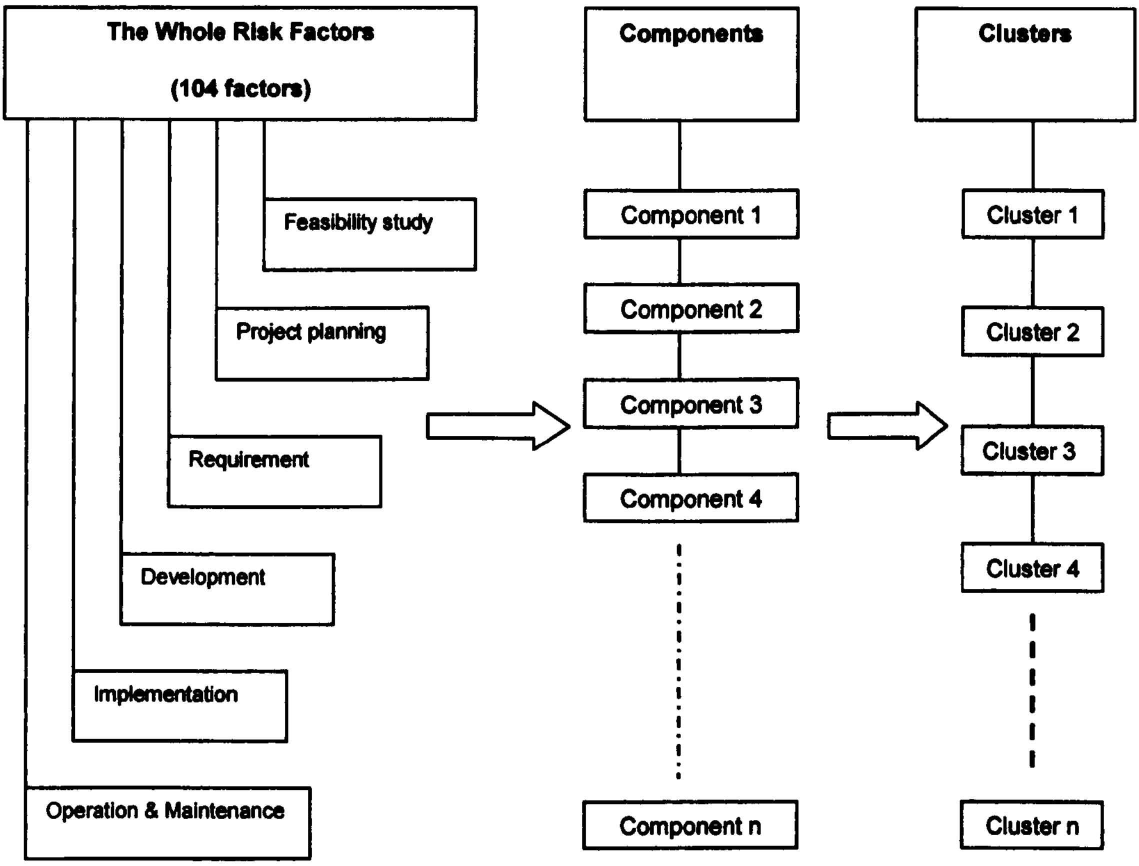


Figure 7.1. The process of data reduction and factor analysis

7.4. Analysis of the Findings

7.4.1. KMO & Bartlett test

Before conducting factor analysis, statistical test using the SPSS was performed to check the possible presence of multicollinearity or correlation among the risk variables; the Kaiser-Meyer-Olkin (KMO) measure for measuring sampling adequacy and the Bartlett Test of Sphericity for testing the presence of correlations (Field, 2005; Morgan et al, 2004). The results are shown in Table 7.1 below:-

Table 7.1 : KMO + Bartlett test

Stages	**Kaiser-Meyer-Olkin (KMO) Measure of Sampling Adequacy**		**Bartlett's Test of Sphericity (Significance value)**	
	Likelihood occurrence	Impact on cost overrun	Likelihood occurrence	Impact on cost overrun
Feasibility study	0.598	0.605	0.000	0.000
Project planning	0.704	0.750	0.000	0.000
Requirement	0.607	0.584	0.000	0.000
Development	0.724	0.742	0.000	0.000
Implementation	0.850	0.687	0.000	0.000
Operation maintenance	0.506	0.505	0.000	0.000

The KMO value varies between 0 and 1. A value of 0 indicates diffusion in the pattern of correlations, which means factor analysis is likely to be inappropriate. Whereas a value of KMO close to 1 indicates the patterns of correlations are relatively compact and factor analysis should yield distinct and reliable factors (Hutcheson & Sofroniou, 1999). Kaiser (1974) recommends accepting values greater than 0.5 as acceptable. Furthermore, values between 0.5 and 0.7 are good, values between 0.7 and 0.8 are great, and values above 0.8 are superb. Table 7.1 (KMO & Bartlett test), shows that the sample is adequate to conduct factor analysis.

Bartlett's test examines whether the population correlation matrix resembles and identity matrix. If the population correlation matrix resembles an identity matrix then it means that every variable correlates very badly with all other variables (i.e; all correlations coefficients are close to zero) (Field, 2005; Morgan et al, 2004). If it were an identity matrix then it would mean that all variables are independent from one another. Given that factor analysis is looking for clusters or components of variables that measure similar things or have common trend and theme, this scenario would not be an ideal situation, as no variables correlate then there are no clusters to find.

Bartlett's measure test the null hypothesis ($H_0 > 0.05$) that the original correlation matrix is an identity matrix. For factor analysis to work, it needs some relationships between variables and the significance value to be ($p < 0.05$). Using a significance level of 0.05, the Bartlett's test shows the values of *p* for both likelihood occurrence and impact on cost overrun are highly significant. This test tells us that the correlation matrix is not an identity matrix; therefore, there are some relationships between the variables. Both the results of KMO and Bartlett test proved that factor analysis is appropriate for these variables.

The results indicate that there is a basis of interpretability that provides sufficient distinctness between project stages within which risk components may be identified and analysed. Moreover, the large sample size used in this study exceeds that normally considered to be adequate for research of this nature (see, for example, Tabachnick and Fidell, 2007; Comrey and Lee, 1992) and at a point at which test parameters become stable irrespective of the participant to variable ratio (Kass & Tinsley, 1979). We have also check the reliability of the discovered risk components using Cronbach's test. The Alpha values of our emerging risks and their impact on cost overrun questions are 0.963 and 0.968 respectively as shown in Table 7.2. This suggests that the discovered risk components have significantly high internal consistency.

Table 7.2. Cronbach alpha test of reliability

	Cronbach Alpha
Likelihood occurrence of risk factors	0.963
Risk factors impact on cost overrun	0.968

7.4.2. Overall Likelihood of occurrence

In Table 7.3, each component is set according to a series of correlations between different risk factors. Thus, it shows how correlated a risk factor could be to other factors. The first column of three sections in Table 7.3, labelled as Initial Eigenvalues related to Eigen value of the correlation matrix and indicates which components of the table remain in analysis. To carry out the factor analysis, only components with Eigen values of more than 1 are selected and those Eigen values of less than 1 are excluded (Punch, 2005; Field, 2005). In the current context, an Eigen value is the amount of the total test variance that is accounted for by a particular factor (Field, 2005; Morgan et al, 2004).

Table 7.3 : Overall Likelihood occurrence

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	31.998	30.767	30.767	31.998	30.767	30.767	29.255	28.030	28.030
2	27.799	26.729	57.496	27.799	26.729	57.496	24.813	23.759	51.789
3	15.440	14.631	72.127	15.440	14.631	72.127	14.152	13.607	65.396
4	12.249	11.951	84.078	12.249	11.951	84.078	11.778	11.391	76.787
5	7.476	7.088	91.166	7.476	7.088	91.166	6.282	6.040	82.827
6	1.993	1.907	93.073	1.993	1.907	93.073	1.834	1.816	84.643
7	1.817	1.647	94.720	1.817	1.647	94.720	1.548	1.489	86.132
8	1.639	1.476	96.196	1.639	1.476	96.196	1.295	1.245	87.377
9	1.426	1.271	97.467	1.426	1.271	97.467	1.114	1.071	88.448
10	0.805	0.374	97.841						
11	0.672	0.246	98.087						
.	.	.	.						
.	.	.	.						
.	.	.	.						
102	-1.287E-14	-1.237E-14	100.000						
103	-1.540E-14	-1.481E-14	100.000						
104	-1.634E-14	-1.571E-14	100.000						

The initial and rotated Eigenvalues were used to confirm the variation explained by each extracted risk component. Lower values indicate that the contribution to the explanation of the variances in the set of our risk survey attributes is minimal.

For example, the initial Eigen value of the first factor in Table 7.3 is 31.998. Hence, the proportion of the total test variance accounted for by the first factor is 28.03% (the figure given in % of variance column). In this analysis for the Overall likelihood of occurrence, just 9 components carry Eigen values of 1 and more, and account for 88.45% of the variance as shown in the cumulative % column. This means that the selected 9 components presents 88.45% of the whole variance. Therefore the 9 components can be considered as the representative of 104 risk factors employed in this study. This means that less than 12% of the existing information is compromised. Another way of presenting the most important factors of a study can be obtained through presentation of a scree plot of data as shown in Figure 7.2.

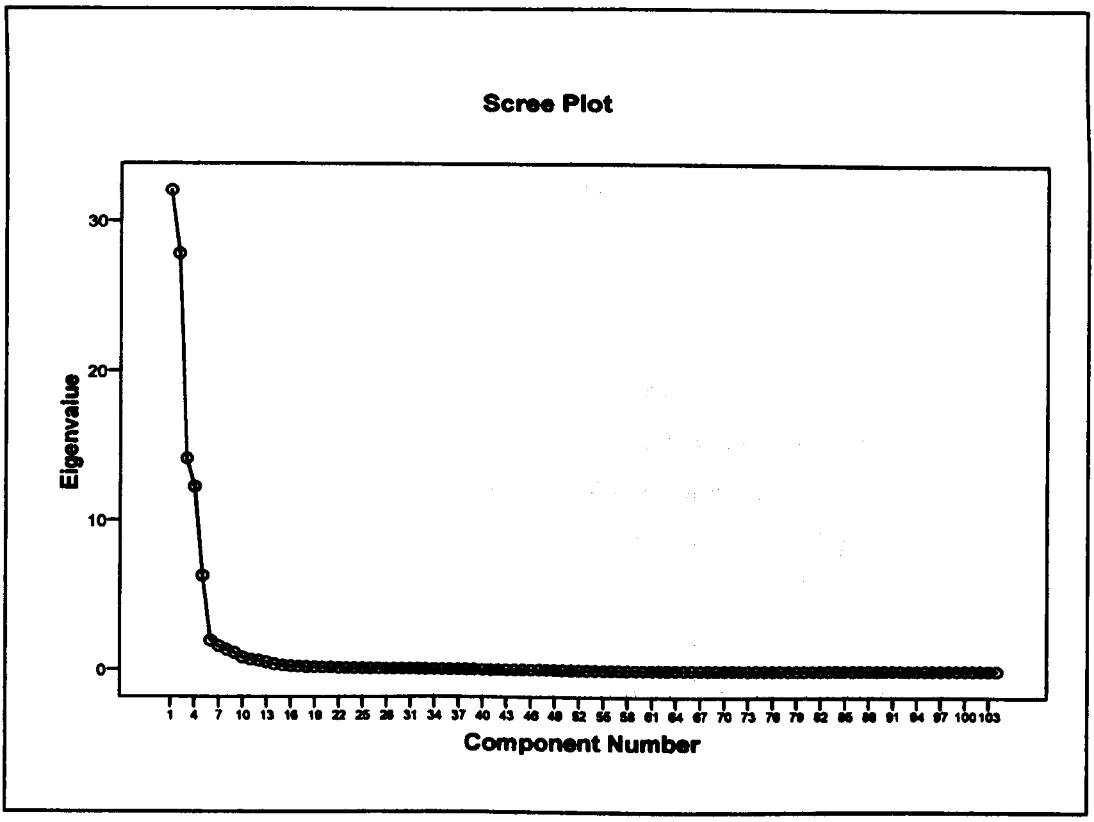


Figure 7.2 : Scree plot of risk factors for Overall likelihood occurrence

The purpose of a scree plot is to provide a graphical picture of the Eigen value for each component extracted in SPSS. As it is shown in the Figure 7.2, the slope of scree is reducing, while moving towards components with Eigen value less than 1. The point of interest is defined between components 9 and 10, where the figure curve connects to the points, starting to flatten out and horizontal. Therefore, in a scree plot, the place where a sharp change in angle occurs can be considered as the exact point that Eigen values of less than 1 are placed (Morgan et al, 2004). On the sharp slope of curve, the Eigen values bigger than one are located, while in the flatten part of the curve, the Eigen values smaller than 1 are plotted.

From principal component analysis 9 components which have the Eigen value of more than 1 are selected. The next phase is the extraction of rotated component matrix for finding out which risk factors are contributing the highest level of influence on the software project. This level of influence is shown in Table 7.4. The Matrix loading score presented in Table 7.4 shows the degree of influence of each risk factor in the whole survey, and the risk factors with the highest rate of influence can be distinguished.

This factor loading tell us about the relative contribution that a variable makes to a factor. Most variables have high loadings on the most important factors, and fewer loadings on other factors. It is recommended to interpret factor loadings with an absolute value greater than 0.4 (ignoring the +ve or –ve sign), which explain around 16% of the variance in the variable. (Stevens, 1995; Maccallum et al, 1999; Morgan et al, 2004; Field, 2005). In Table 7.4, only the degree of influence 0.4 and above are shown. However, for the purpose of interpretation of how to extract the risk factors based on the factor loading, the factor loading of the components for some of risk factors were shown in Table 7.4. As for the Implementation stage (factor IM1-IM24) and Operation and maintenance stage (factor OP1-OP19), the factor loading were not shown as the loading score were all below 0.4, hence the risk factor from this two stages were not selected for the components.

For example, from Table 7.4, the risk factor (F1; 0.656) has got greater influence on component 7 compared to other components. Whereas, the risk factor (P5; 0.509) has got more influence on component 6 in relation to other components, and (R3; 0.526) has got more influence on component 4 in relation to other components. As can be seen from the Table 7.4, for risk factors D18-D20 and D24-D35, the factor loading scores were below 0.4, so were not selected. This same method is used for the rest of the risk factors and components to extract the most effective risk factors for each component. The risk factors with the high scores and correlation values are chosen for each component.

Table 7.4 : Matrix loading score for Overall Likelihood occurrence

	Component								
	1	2	3	4	5	6	7	8	9
F1		0.175	-0.156			0.17	**0.656**	-0.166	
F2		**0.576**			-0.165			0.145	0.237
F3	-0.274	0.208	0.117			**0.637**		-0.131	
F4			-0.184	0.296	0.288	-0.182			**0.495**
F5		**0.561**	0.178	-0.238		0.257	0.185		0.126
P1	0.146		-0.141	0.178	-0.157	**0.457**			-0.211
P2		0.212	0.137		0.194		-0.221	**0.532**	0.234
P3							0.278	**0.722**	0.213
P4		-0.223			-0.269	0.12		**0.78**	
P5	0.234		0.171		0.105	**0.509**	0.291	-0.165	0.245
P6	0.197		0.181	-0.217	0.268	**0.418**	0.279	0.189	-0.138
P7	-0.149	-0.265	0.152	0.181	0.209	**0.443**		0.248	0.154
P8							0.137	**0.875**	-0.137
P9		0.286	0.123	0.192	**0.62**	0.274	0.285	0.133	-0.241
P10		0.143	**0.807**		0.15		-0.145	-0.114	0.102
P11	0.123	-0.156	-0.144			**0.848**		0.276	0.102
P12					**0.59**	0.21		0.2	
P13			-0.175		**0.892**		-0.259	0.15	0.142
P14			-0.258	0.121		**0.676**			-0.158
R1				**0.803**					
R2				**0.553**					
R3				**0.526**					

R4				0.456					
R5				0.656					
R6				0.572					
R7	0.551								
D1		0.827							
D2		0.493							
D3		0.708							
D4		0.856							
D5		0.688							
D6		0.568							
D7		0.691							
D8		0.474							
D9		0.611							
D10		0.835							
D11	0.55								
D12		0.427							
D13					0.808				
D14								0.477	
D15	0.448								
D16	0.097	-0.314	0.337	-0.305	0.002	0.027	0.204	0.004	-0.224
D17	0.414								
D18	0.358	-0.059	0.113	0.073	0.043	0.328	-0.305	0.179	0.2
D19	-0.061	0.385	0.207	0.295	0.082	-0.123	0.016	0.239	-0.066
D20	0.081	-0.396	0.3	-0.31	0.191	0.095	0.033	-0.343	0.063
D21			0.551						
D22			0.592						
D23			0.692						
D24	-0.194	0.054	0.272	-0.264	0.037	0.214	0.001	-0.187	0.099
D25	0.094	0.002	-0.242	0.163	0.071	-0.05	0.211	0.073	-0.139
D26	0.031	-0.132	0.225	0.104	-0.002	0.164	0.286	0.131	0.008
D27	-0.058	0.089	0.207	-0.309	0.256	0.09	-0.182	-0.05	0.161
D28	0.351	0.045	0.084	0.31	0.229	0.336	0.171	0.202	-0.075
D29	-0.112	0.176	0.092	0.188	0.002	0.227	0.151	0.068	0.075
D30	0.134	0.306	-0.211	0.347	0.079	-0.065	0.12	-0.189	0.16
D31	0.084	-0.103	0.329	-0.067	0.069	0.062	0.103	0.02	-0.161
D32	0.106	0.156	0.163	0.049	0.243	-0.16	-0.058	-0.174	0.094
D33	0.067	-0.058	0.037	-0.246	0.145	0.109	0.363	0.022	-0.137
D34	-0.137	0.072	0.124	0.012	0.192	0.203	0.161	0.117	0.039
D35	0.045	-0.281	0.026	0.199	-0.294	0.119	0.215	-0.103	0.034
IM1 . . . IM24	The factor loading score for the Implementation stage (risk factor IM1 – IM24), was not shown, as they were all below 0.4, hence the risk factor in this stage were not selected for the components.								
OP1 . . . OP19	The factor loading score for the Operation and maintenance stage (risk factor OP1 – OP19), was not shown, as they were all below 0.4, hence the risk factor in this stage were not selected for the components.								

The most important and influential risk factors of each component are extracted to form a reduced list of risk factors, which are highly manageable without losing a large amount of data. By applying factor analysis and data reduction in this survey, the questionnaire 104 risk factors are reduced to 9 components which are shown in Table 7.5 below. The percentages of variance of each component in Table 7.5 are extracted from Table 7.3. Common themes of the components were identified and each component was given new terms for reference.

Table 7.5 : Components for the Overall Likelihood of occurrence

Risk component	Extracted eigenvalue	Extracted sum of squared loadings: variance %	Rotation sum of squared loadings: variance %	Risk factors aggregated to component following rotation	
				Factor loading score	
Component 1 Project user engagement	31.99	30.76	28.03	0.551 0.550 0.448 0.414	Lack of users involvement in requirement stage: R7 Failure of user acceptance test: D11 Lack of users involvement and commitment: D15 Ineffective communication within development team members: D17
Component 2 Technology failure	27.79	26.73	23.76	0.576 0.561 0.827 0.493 0.708 0.856 0.688 0.568 0.691 0.474 0.611 0.835 0.427	Too narrow focus on the technical IT issues: F2 Inappropriate technology chosen from the feasibility study: F5 Inproper handover from the requirement team: D1 Inappropriate development methodology used: D2 Unsuitable working model and prototype: D3 Programming language and CASE tool selected not adequate: D4 High level of technical complexities: D5 Project involves the use of new technology: D6 Difficulty in defining the input and output of system: D7 Immature technology: D8 Technological advancements and changes: D9 Failures and inconsistencies of unit/modules test results: D10 Time consuming for testing: D12
Component 3 Project personnel	15.44	14.63	14.15	0.807 0.551 0.592 0.692	Project management & development team not properly set up: P10 Inexperienced team members: D21 Lack of commitment to project among development team members: D22 Ineffective and inexperienced project manager: D23
Component 4 Technology and system requirements	12.25	11.95	11.39	0.803 0.553 0.526 0.456 0.656 0.572	Unclear and inadequate identification of systems requirements: R1 Incorrect systems requirements: R2 Misinterpretations of the systems requirements: R3 Conflicting system requirements: R4 Gold plating or unnecessary functions and requirements: R5 Inadequate validation of the requirements: R6

The table identifies risk components which are groupings of risk factors from the 104 initially identified. Factor analysis is employed and risk components with eigenvalues in excess of 1 are extracted, leaving a total of 9. The table reports both the variance explained by these retained factors from the total variance of all 104 factors as well as the factor loadings (and their variances) following varimax rotation (an orthogonal rotation method) in which the variance of each of the factors is maximised. This facilitates interpretability of the resulting factors. The retained risk factors (total, 45) and their groupings into risk components are shown in the final column. The initial identifiers (R, D, F and P) in the final column – followed by a number – refer to the project stages and their initial number within each stage. Thus, the stages are: F- Feasibility Study; D – Development; R –Requirement; P – Project Planning.

Risk component	Extracted eigenvalue	Extracted sum of squared loadings: variance %	Rotation sum of squared loadings: variance %	Risk factors aggregated to component following rotation	
				Factor loading score	
Component 5 Project implementation	7.48	7.09	6.28	0.620 0.590 0.892 0.808	Critical and non-critical activities of project not identified: P9 Lack of contingency plan/back up: P12 System conversion method not well planned: P13 Resources shifted from project during development: D13
Component 6 Project planning	1.99	1.91	1.82	0.637 0.457 0.509 0.418 0.443 0.848 0.676	Overlooked the management and business impact issues: F3 Unclear project scope and objectives: P1 Improper change management planning: P5 Inaccurate estimate of resources: P6 Unrealistic project schedule: P7 Unclear line of decision making authority throughout the project: P11 Improper planning of timeframe for project reviews and updating: P14
Component 7 Feasibility study	1.82	1.65	1.49	0.656	Wrong justifications of cost benefit analysis from feasibility study: F1
Component 8 Project process	1.64	1.48	1.24	0.532 0.722 0.780 0.875 0.477	Undefined project success criteria: P2 Lack of quality control procedure and mechanism: P3 Project milestones for stages not well established: P4 Inadequate detail work breakdown structure: P8 Changes in management of organisation during development: D14
Component 9 Feasibility study decision	1.43	1.27	1.07	0.495	Wrong justification of investment alternatives and opportunity cost: F4

The table identifies risk components which are groupings of risk factors from the 104 initially identified. Factor analysis is employed and risk components with eigenvalues in excess of 1 are extracted, leaving a total of 9. The table reports both the variance explained by these retained factors from the total variance of all 104 factors as well as the factor loadings (and their variances) following varimax rotation (an orthogonal rotation method) in which the variance of each of the factors is maximised. This facilitates interpretability of the resulting factors. The retained risk factors (total, 45) and their groupings into risk components are shown in the final column. The initial identifiers (R, D, F and P) in the final column – followed by a number – refer to the project stages and their initial number within each stage. Thus, the stages are: F- Feasibility Study; D – Development; R –Requirement; P – Project Planning.

7.4.3. Impact of risk factors on cost overrun

As the interpretation of the Eigen values correlation matrix to the factor analysis already being explained in the previous section (for Likelihood occurrence), these next few sections will explained only the most relevant data for the impact of risk factors on cost overrun.

Table 7.6 : Total Variance Explained – Impact of risk factors on Cost overrun

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	32.202	30.963	30.963	32.202	30.963	30.963	28.333	27.143	27.143
2	21.481	20.578	51.541	21.481	20.578	51.541	19.625	18.809	49.952
3	16.762	16.079	67.620	16.762	16.079	67.620	15.727	15.022	60.974
4	14.385	13.694	81.314	14.385	13.694	81.314	12.662	12.175	73.149
5	12.290	11.602	92.916	12.290	11.602	92.916	10.793	9.493	82.642
6	3.817	3.661	96.577	3.817	3.661	96.577	3.609	3.559	86.201
7	2.319	2.129	98.706	2.319	2.129	98.706	1.791	1.722	87.923
8	.217	.181	98.887						
9	.156	.131	99.018						
.	.	.	.						
.	.	.	.						
.	.	.	.						
102	-1.287E-14	-1.237E-14	100.000						
103	-1.540E-14	-1.481E-14	100.000						
104	-1.634E-14	-1.571E-14	100.000						

For Impact of risk factors on cost overrun, the Eigen value of the first factor in Table 7.6, is 32.202. Hence, the proportion of the total test variance accounted for by the first factor is 30.963% (the figure given in % of variance column). In this analysis for Impact of risk factors on cost overrun, just 7 components carry Eigen values of 1 and more, and account for 87.92% of the variance as shown in the cumulative % column. This means that the selected 7 components presents 87.92% of the whole variance for the Impact of risk factors on cost overrun and less than 13% of original data is compromised.

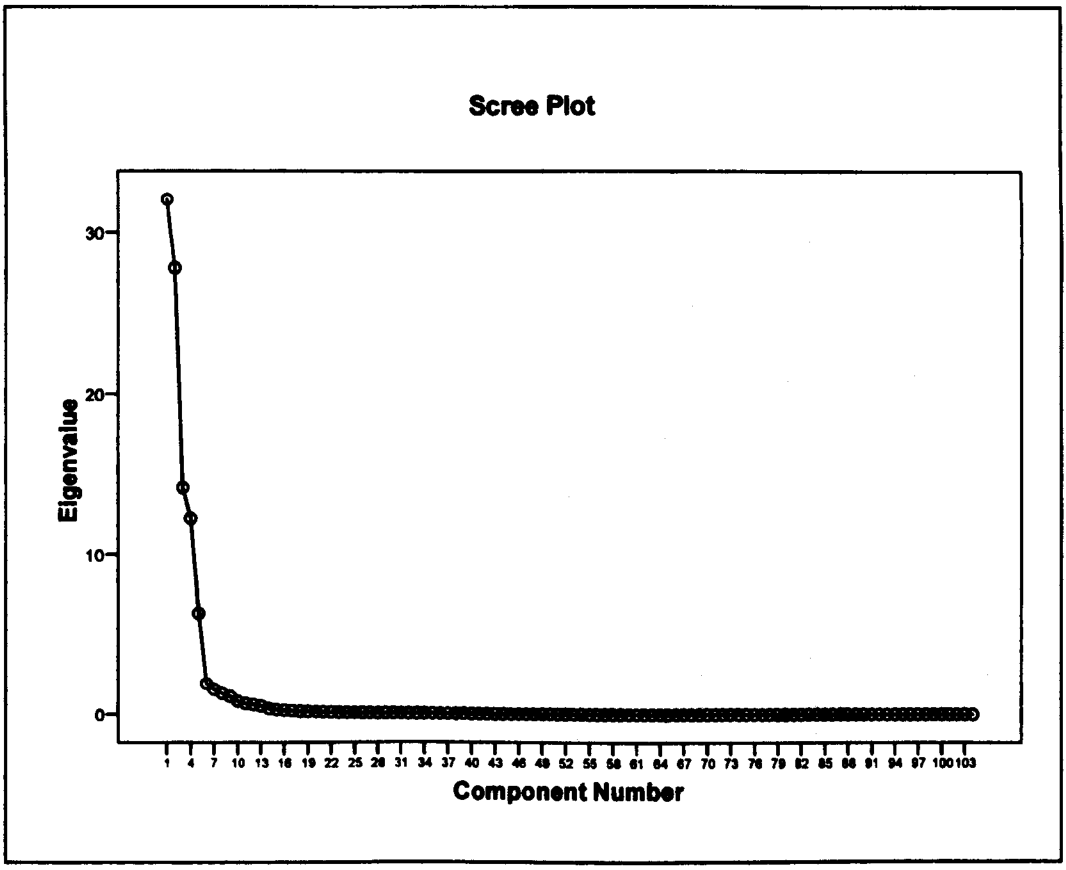


Figure 7.3 : Scree plot for Impact of risk factors on cost overrun

The point of interest in the scree plot in Figure 7.3 for Impact of risk factors on cost overrun is defined between components 7 and 8, where the figure curve connects to the points, starting to flatten out and horizontal.

Table 7.7 presented shows the degree of influence of each risk factor in the survey for Impact of risk factors on cost overrun, and the risk factors with the highest rate of influence and high loading matrix can be distinguished. The same method and concept of factor loading score used previously with the likelihood occurrence, is also used for the risk factors impact on cost overrun in extracting the most important risk factors.

Table 7.7 : Matrix loading score for Impact of risk factors on cost overrun

	Component						
	1	2	3	4	5	6	7
F1				0.19	**0.663**		0.169
F2		**0.65**		-0.128		0.176	0.298
F3	**0.536**	0.167	-0.109	0.154	0.174		
F4		-0.257			**0.619**	-0.176	0.185
F5	-0.125	**0.623**	-0.248				-0.12
P1	**0.567**				0.127	-0.122	
P2	0.298		0.149				**0.843**
P3	-0.19	-0.115		0.282	-0.191		**0.827**
P4	0.178				0.144		**0.813**
P5	-0.189			-0.112		**0.407**	
P6	**0.577**	-0.156	0.102	0.3	0.118	-0.227	
P7		0.191	0.243			**0.637**	-0.146
P8		0.111		0.103	-0.191	0.262	**0.842**
P9					0.125	-0.232	**0.687**
P10	-0.236		0.223	-0.264		0.162	**0.84**
P11				0.254	-0.149	**0.654**	-0.103
P12	0.235		-0.278	-0.249		**0.448**	0.172
P13				**0.773**	0.159		
P14	**0.482**	-0.151	0.222	0.269			-0.131
R1			**0.629**			0.178	
R2	-0.258	0.297	**0.693**				
R3			**0.639**	-0.26		-0.112	0.275
R4		0.136	**0.767**			0.189	
R5		0.101	**0.836**	0.257	0.203	0.156	0.174
R6			**0.66**		0.24		0.275
R7		**0.646**	**0.782**	-0.173	0.203	-0.123	
D1						**0.602**	
D2				**0.467**		-0.176	0.25
D3		-0.235		**0.729**	0.176		
D4			0.114	**0.717**	-0.263		-0.167
D5		**0.727**	-0.171				
D6		**0.662**	0.129		-0.122	0.209	
D7		-0.107		**0.677**	0.228		
D8	-0.173	**0.747**		0.254	-0.221	0.208	0.27
D9	-0.134	**0.82**	0.149	-0.258		0.127	
D10		-0.235		**0.729**	-0.176		
D11				**0.72**	0.171		-0.167
D12		-0.121		**0.897**		0.16	
D13	-0.245				-0.166	**0.499**	0.196
D14	0.285			0.28	-0.214	**0.476**	
D15	**0.661**						0.25
D16	0.047	0.339	-0.068	-0.043	0.05	0.017	0.155
D17	0.24	0.157		0.259	0.284	**0.666**	0.149
D18	-0.138	0.176	0.094	0.002	0.297	-0.335	0.138
D19	0.182		-0.164	-0.16		0.147	**0.774**
D20	-0.025	-0.354	0.163	0.355	0.15	-0.014	0.386
D21	0.136	-0.303	0.373	0.168	-0.105	0.173	-0.035
D22	**0.436**	0.14	-0.282			-0.29	-0.265

D23	0.776	0.24	-0.295	0.19	0.068	-0.077	0.244
D24	0.078	0.085	0.084	0.121	0.159	0.256	-0.131
D25	-0.04	0.278	-0.276	-0.128	-0.296	0.131	0.202
D26	0.079	-0.183	0.205	0.324	0.351	-0.088	0.318
D27	0.052	0.12	-0.038	-0.022	-0.18	0.185	-0.242
D28	-0.309	0.097	0.183	0.092	0.224	-0.113	0.303
D29	0.029	-0.095	-0.111	-0.009	0.182	0.034	-0.023
D30	-0.082	0.332	0.272	0.139	-0.117	-0.096	0.27
D31	0.096	-0.254	-0.061	-0.21	0.089	0.076	-0.384
D32	-0.379	-0.123	0.194	0.128	0.201	0.124	0.041
D33	0.081	0.333	0.1	-0.14	0.12	-0.269	-0.271
D34	-0.3	0.183	-0.003	0.25	-0.001	0.084	0.069
D35	0.31	-0.164	0.203	0.216	-0.049	-0.344	0.12
IM1 . . . . IM24	The factor loading score for the Implementation stage (risk factor IM1 – IM24), was not shown, as they were all below 0.4, hence the risk factor in this stage were not selected for the components.						
OP1 . . . . OP19	The factor loading score for the Operation and maintenance stage (risk factor OP1 – OP19), was not shown, as they were all below 0.4, hence the risk factor in this stage were not selected for the components.						

The result of this extraction is presented in Table 7.8. The most important and influential risk factors of each component are extracted to form a reduced list of risk factors, which are highly manageable without losing a large amount of data. By applying factor analysis and data reduction in this survey, the questionnaire 104 risk factors are reduced to 7 components which are shown in Table 7.8 below. The percentages of variance of each component in Table 7.8 are extracted from Table 7.6.

Table 7.8 : Components for the Impact on risk factor on cost overrun

Risk component	Extracted eigenvalue	Extracted sum of squared loadings: variance %	Rotation sum of squared loadings: variance %	Risk factors aggregated to component following rotation	
				Loading factor	
Component 1 Project team planning	32.202	30.963	27.143	0.535	Overlooked the management and business impact issues: F3
				0.557	Unclear project scope + objectives: P1
				0.577	Inaccurate estimate of resources: P6
				0.482	Improper planning of timeframe for project reviews and updating: P14
				0.561	Lack of users involvement and commitment: D15
				0.436	Lack of commitment among development team members: D22
				0.776	Ineffective and inexperienced project manager: D23
Component 2 Technology appropriateness	21.481	20.578	18.809	0.650	Too narrow focus on the technical IT issues: F2
				0.623	Inappropriate technology chosen from the feasibility study: F5
				0.727	High level of technical complexities: D5
				0.662	Project involves the use of new technology: D6
				0.747	Immature technology: D8
				0.820	Technological advancements and changes: D9
Component 3 Technology specification	16.762	16.079	15.022	0.529	Unclear and inadequate identification of systems requirements: R1
				0.593	Incorrect systems requirements: R2
				0.539	Misinterpretations of the systems requirements: R3
				0.757	Conflicting system requirements: R4
				0.836	Gold platting or unnecessary functions and requirements: R5
				0.560	Inadequate validation of the requirements: R6
				0.782	Lack of users involvement in requirement stage: R7
Component 4 Technology and implementation	14.385	13.694	12.175	0.773	System conversion method not well planned: P13
				0.467	Inappropriate development methodology used: D2
				0.729	Unsuitable working model and prototype: D3
				0.717	Programming language and CASE tool selected not adequate: D4
				0.577	Difficulty in defining the input and output of system: D7
				0.729	Failures and inconsistencies of unit/modules test results: D10
				0.720	Failure of user acceptance test: D11
				0.897	Time consuming for testing: D12
Component 5 Feasibility study	12.290	11.602	9.493	0.553	Wrong justification of cost benefit analysis from feasibility study: F1
				0.619	Wrong justification of investment alternatives and opportunity cost: F4
Component 6 Project team management	3.817	3.661	3.559	0.407	Improper change management planning: P5
				0.637	Unrealistic project schedule: P7
				0.654	Unclear line of decision making authority throughout the project: P11
				0.448	Lack of contingency plan/back up: P12
				0.502	Inproper handover from the requirement team: D1
				0.499	Resources shifted from project during development: D13
				0.475	Change in management during development: D14
				0.666	Ineffective communications within development team members: D17
Component 7 Project team activities	2.319	2.129	1.722	0.843	Undefined project success criteria: P2
				0.827	Lack of quality control procedure and mechanism: P3
				0.813	Project milestones for stages not well establish: P4
				0.842	Inadequate detail breakdown structure: P8
				0.587	Critical and non-critical activities of project not identified: P9
				0.774	Inadequately trained development team members: D19

7.4.4. Geographical spread of respondents

The geographical spread of respondents, whilst from a wider base than nearly every other study is, nonetheless dominated by the USA and UK which accords with previous research samples. Thus, the survey respondents reflect geographical sources of other studies but is extended geographically. No other country contains more than 10% of the survey sample and hence do not expect any geographical skewness in the results that is different from studies focussing on the USA and UK. Of the studies reported during the literature reviews that were use to compare samples the following sample survey geographical characteristics were reported; Chang 2006, not specified, but likely to be USA given the reporting of the results; Keil et al 2002, USA only; Liu et al 2009, China only; Wallace et al 2004, unspecified.

Nonetheless, an ANOVA analysis was conducted to test the difference means between geographical groups. The results from ANOVA analysis for the Likelihood occurrence shows that at a significant level of (0.05), there is statistical significance of the respondents by countries for Component 2, Component 3 and Component 8. In other words, at 5% confidence level for Component 2, Component 3 and Component 8, the significance value of less than 0.05 for these 3 components implies that, less than 5% of this resulted by chance for the whole population.

Whereas the ANOVA results for the impact on cost overrun indicate that at a significant level of (0.05), there is statistical significance of the respondents by countries for Component 2 and Component 6. At 5% confidence level for Component 2 and Component 6, the significance value of less than 0.05 for these two components implies that, less than 5% of this resulted by chance for the whole population. These two ANOVA results demonstrate that our sample is representative of the general population for the purpose of validating our instrument and assessing generalizability of our model. As a consequence, it is felt that the respondent sample reported here is broadly comparable with other research in this area.

7.5. Discussion

7.5.1. Clustering of Likelihood occurrence of risk factors

This initial clustering into 9 components and the relationship to individual risk factors is entirely empirically determined, in common with previous research adopting this approach (for example, Ropponen and Lyytinen, 2006). The factor loadings following rotation indicate a shift in importance of individual factors to the risk components and redistribution in the overall explanation with the total variance accounted for is marginally reduced to 88.45%. The research adopt a varimax rotation that maintains the orthogonality of the individual factors and potentially enhances their interpretability. The major components (those with the largest variances and with variances reported in parenthesis) are: 1 28.03%), 2 (23.76%), 3 (14.15%) and 4 (11.39%). The identification of risk factors in Table 7.4 provides a guide to the interpretation of the risk components and articulating the findings and to present a view on clustering in the context of the whole-life cycle project to determine risks in relation to meaningful project stages.

Thus, the 9 risk components being assessed and interpreted what clusters could be formed that might be placed into a whole-life cycle context. This approach follows that of Wallace and Keil (2004) who employ socio-technical systems theory to help establish the dimensionality of risks they observe in their own survey (n=507). The approach is to allow factor analysis to establish the initial dimensionality (the 9 risk components of Table 7.4) and then to interpret the results in terms of the project whole-life cycle. This latter element is akin to that employed in Barki et al (1993). In examining the make-up of the risk components (that is, from the risk factors in the Table 7.4) a number of themes were observed to be consistent with categorisations from a generalised project plan over the whole-life cycle. This is supported by an examination of the factor loadings of the risk factors which are reported in the fifth column of Table 7.5 and which reports the loading factors extracted from the rotated component matrix of the risk data sample. This is the main basis for the component interpretation used.

From the analysis, only 45 risk factors out of the 104 initially surveyed was selected which account for 88.45% of the variance that could be explained. The 45 risk factors were selected based on the *'eigen-one'*, Kaiser (1960) criterion cut-off, which used only risk factors that have factor loading of 0.400 and above. The 9 components extracted from factor analysis were then clustered together to form a few clusters that have some common themes. Each.cluster degree of effect in the likelihood occurrence of risk is calculated based on percentage of variance of each component derived from Table 7.3.

The analysis shows 3 main clusters emerge that are consistent and which we label:-

- Cluster 1: Feasibility study.
- Cluster 2: Project and team management.
- Cluster 3: Technology requirement.

The new clusters for the likelihood occurrence of risk factors are shown in Table 7.9 below :-

Table 7.9 : New cluster for likelihood occurrence of risk

Cluster 1	Component	% variance	Main risk factors	Total % variance
Cluster 1 Feasibility study	Component 7	1.49	• F1: Wrong justifications of cost benefit analysis from feasibility study.	2.56 %
	Component 9	1.07	• F4 : Wrong justification of investment alternatives and opportunity cost	
Cluster 2 Project and team management	Component 1	28.03	• R7: Lack of users involvement in requirement stage • D11: Failure of user acceptance test • D15: Lack of users involvement and commitment • D17: Ineffective communication within development team members	51.52 %
	Component 3	14.15	• P10: Project management & development team not properly set up • D21: Inexperienced team members • D22: Lack of commitment to project among development team members • D23: Ineffective and inexperienced project manager	
	Component 5	6.28	• P9: Critical and non-critical activities of project not identified • P12: Lack of contingency plan/back up • P13: System conversion method not well planned • D13: Resources shifted from project during development	
	Component 6	1.82	• F3: Overlooked the management and business impact issues • P1: Unclear project scope and objectives • P5: Improper change management planning • P6: Inaccurate estimate of resources • P7: Unrealistic project schedule • P11: Unclear line of decision making authority throughout the project • P14: Improper planning of timeframe for project reviews and updating	
	Component 8	1.24	• P2: Undefined project success criteria • P3: Lack of quality control procedure and mechanism • P4: Project milestones for stages not well established • P8: Inadequate detail work breakdown structure • D14: Changes in management of organisation during development	
Cluster 3 Technology requirement	Component 2	23.76	• F2: Too narrow focus on the technical IT issues • F5: Inappropriate technology chosen from the feasibility study • D1: Improper handover from the requirement team • D2: Inappropriate development methodology used • D3: Unsuitable working model and prototype • D4: Programming language and CASE tool selected not adequate • D5: High level of technical complexities • D6: Project involves the use of new technologies • D7: Difficulty in defining the input and output of system • D8: Immature technology • D9: Technological advancements and changes • D10: Failures and inconsistencies of unit/modules test results • D12: Time consuming for testing	35.15 %
	Component 4	11.39	• R1: Unclear and inadequate identification of systems requirements • R2: Incorrect systems requirements • R3: Misinterpretations of the systems requirements • R4: Conflicting system requirements • R5: Gold plating or unnecessary functions and requirements • R6: Inadequate validation of the requirements	

Cluster 1 comprises components 7 and 9 and represents 2.56% of the total variance explained. Only 2 risk factors make-up this cluster and they relate to cost benefit analysis and an analysis of opportunity costs in the initial evaluation. It is perhaps not surprising that these factors could be related in this manner since a full perspective of the cost and benefits would seek to avoid the risks associated with both incorrect conclusions from the analysis undertaken and a subsequent failure to incorporate all relevant factors. From the perspective of IT professionals, it could be argued that any failure in assessment at the project feasibility stage might manifest as project problems later-on. This would, as a consequence, have an impact on project success. The expectation is that IT professionals would prioritise this issue, if only to avoid dealing with the consequences of a situation not of their making later in the project life cycle.

However, the issue could become problematic if they are not, as is possible, involved in the project at this stage. Furthermore, the identification of opportunity costs requires a deep knowledge of the organisational context in order to be able to allocate them successfully. Invariably, this will need, and at an early stage, accurate projections of cash flows including the opportunity costs of capital (Ballantine & Stray, 1998). The fact that IT professionals highlight this as a significant weight in the survey response might be reflective of their lack of oversight and detailed knowledge of organisational context which would provide an appropriate framework to judge what may or may not be appropriate feasibility risks to identify and evaluate.

What is clear from this cluster is the potential for organisations to embed problems in the project due to poor interaction and communication between managers within the organisation and those external consultants who will be charged with managing the project later in the timeline. Ultimately, this is an issue of the assumptions held by the various parties involved in the decision making and the relative boundaries of consideration they hold relative to the potential for embedding risk in the project.

Cluster 2 comprises components 1, 3, 5, 6 and 8 and represents 51.52% of the total variance explained. This has been labelled so as to include issues of Project and Team Management. The basis of this, in looking at the associated risk factors in Table 7.3, is indicated by a range of factors concerning the interaction of the project with non-technical software areas, and particularly regarding interaction with staff and other firm resources. Again, the issues of communication and information sharing could be seen to play an important role in shaping this problem. Whilst there is one factor (*Lack of user involvement in the requirement stage*) that may overlap with the Feasibility study cluster, the remaining factors relate to the immediacy of project implementation.

Components 1 and 3 relate to project interaction with resources available and Components 5, 6 and 8 relate to the consequences of failure at the planning stage. Thus, on the resource side the observe risk factors were ranging from lack of human input, to inexperience, lack of commitment, mid-project resource-shifting, inadequate line management, along with overlooked business impact issues where the resources interaction with the wider organisational context is evident. With respect to the consequences of planning failure, a full spectrum of risks were observed that were inadequately anticipated but specifically contextualised in terms of project performance. The risk factors are often placed in the past tense and responses are provided with the benefit of hindsight and with the project running or even complete. It might be right to question why these *ex post* factors do not appear as planning issues specifically. As such, they could be regarded as risk factors that naturally pre-suppose a limit on the ability of actors to forecast risk at the feasibility stage which arguably validates our division of risk components along project cycle criteria.

It could be argued that Cluster 2 is really an updating and incorporates experiences of risks relating to earlier stages and also of new risks uniquely related to project implementation. On an aspect noted earlier, it was mentioned that no risk factors relating to *Implementation* and *Operation and Maintenance* survived the cut-off of the extracted factors. This now seems not to be an oversight on the part of IT professionals, but a perspective on risks that they could not agree on since they did not emerge as factors exhibiting sufficient correlations to factorise. In interpreting why this might be the case, it is possible that of those risk factors identified in Cluster 2, the risk factors observed were the consequences – broadly defined – of the risks relating to *Implementation* and *Operation and Maintenance* as they begin to become evident during project build and completion. In this respect, it seems that the wider context of risks relating to 'Project and Team Management', as was labelled Cluster 2, more accurately reflect the risk perspective of IT professionals of risk factors identified at some distance from the organisational detail and context.

Cluster 3 is defined as 'Technology requirement' and this is comprised of components 2 and 4 and represents 35.15% of the total variance explained. The range of risks in this cluster anticipate a wide range of problems but, as with Cluster 2, they appear to involve both the crystallisation of risks not earlier anticipated in Cluster 1 and of new risks emerging that relate to the inadequacies of various aspects of technology as they first become operational. With respect to failure of planning it was noted that inappropriate choice of technology at the feasibility stage as an identified factor in this cluster along with unclear and inadequate identification of systems requirements and even incorrect systems requirements. Of the risks that are likely to appear only once the project is partly implemented, at least, the underperformance of technology as a dominant feature.

This manifests in terms of narrow focus; inadequacy of development methodologies, programming languages, and working models, and programme or module failure; the use of new, immature, highly complex or even outdated technology; misinterpretation of systems requirements and conflicting systems requirements; and the risks relating to project testing, specifically that of extended time periods and inadequate validation. As before, it was noted that there is a build-up of risks from failures at earlier stages combined with risks that are unique to Technology that could only become apparent at a point when some part of the project has been implemented. This cluster also is seen as the technology context for *Implementation* and *Operation and Maintenance* risk factors that did not earlier survive the cut-off. This would seem natural, as mention earlier, given the perspective of IT professionals.

7.5.2. Clustering of Impact of risk factors on cost overrun

Table 7.6 details the results relating to the factor analysis of risks relating to their impact cost overrun. Again, only factors with eigenvalues of more than 1 are retained and, on this basis, 7 risk components are identified. Using the same concept of clustering of the Likelihood occurrence of risk previously explained, only 45 risk factors of the impact on cost overrun out of the 104 initially surveyed was selected which account for 87.92% of the variance that could be explained. The 45 risk factors were selected based on the *'eigen-one'*, Kaiser (1960) criterion cut-off, which used only risk factors that have factor loading of 0.400 and above. The major components (those with the largest variances and with variances reported in parenthesis) are: 1 (27.14%), 2 (18.81%), 3 (15.02%) and 4 (12.18%).

Given the research earlier discussion concerning the role of cost in project IT risk analysis, it is no surprise that the interpretability and mapping of risk onto a cost view of the project life cycle extracts similar risk factors, although their clustering is somewhat different. This can be considered to be an early validation of the cost perspective as a relevant and distinct partition of risk. In looking for themes with which to analyse the components and their factor content in the context of the project life cycle, project clusters were formed from the identified components and their rotated risk factors. Three clusters are identified and, as before, labelled them *Cluster 1*: Feasibility study; *Cluster 2*: Project Team Management; and *Cluster 3:* Technology Requirement.

Cluster 1 comprises component 5 only and represents 9.49% of the explained variance. The same risk factors are identified as before in the likelihood occurrence and with an improved level of explanation (variance of 2.56% previously reported for the likelihood occurrence). Component 5 loads positively on risk factors F1 and F4. One view, which the research tentatively offer, is that risk relating to an adequate feasibility study is likely to be interpretable fully in terms of a cost outcome. Moreover, given the proportion of the variance explained (from the rotated factors), it is clear that IT professionals judge this stage of the project life cycle as critical. This suggests that it is possible to take a view of the success of a project, and its risk factors, from an early stage in respect of both their likelihood of occurrence and of their impact on the potential for cost overrun.

Table 7.10 : New clusters for impact of risk factors on cost overrun

Cluster 1	Component	% variance	Main risk factors	Total % variance
Cluster 1 **Feasibility study**	Component 5	9.493	• F1: Wrong justification of cost benefits analysis from feasibility study. • F4: Wrong justification of investment alternatives and opportunity cost.	9.493
Cluster 2 **Project team management**	Component 1	27.143	• Overlooked the management and business impact issues: F3 • Unclear project scope + objectives: P1 • Inaccurate estimate of resources: P6 • Improper planning of timeframe for project reviews and updating: P14 • Lack of users involvement and commitment: D15 • Lack of commitment among development team members: D22 • Ineffective and inexperienced project manager: D23	32.424
	Component 6	3.559	• Improper change management planning: P5 • Unrealistic project schedule: P7 • Unclear line of decision making authority throughout the project: P11 • Lack of contingency plan/back up: P12 • Inproper handover from the requirement team: D1 • Resources shifted from project during development: D13 • Change in management during development: D14 • Ineffective communications within development team members: D17	
	Component 7	1.722	• Undefined project success criteria: P2 • Lack of quality control procedure and mechanism: P3 • Project milestones for stages not well establish: P4 • Inadequate detail breakdown structure: P8 • Critical and non-critical activities of project not identified: P9 • Inadequately trained development team members: D19	
Cluster 3 **Technology requirement**	Component 2	18.809	• Too narrow focus on the technical IT issues: F2 • Inappropriate technology chosen from the feasibility study: F5 • High level of technical complexities: D5 • Project involves the use of new technology: D6 • Immature technology: D8 • Technological advancements and changes: D9	46.006
	Component 3	15.022	• Unclear and inadequate identification of systems requirements: R1 • Incorrect systems requirements: R2 • Misinterpretations of the systems requirements: R3 • Conflicting system requirements: R4 • Gold platting or unnecessary functions and requirements: R5 • Inadequate validation of the requirements: R6 • Lack of users involvement in requirement stage: R7	
	Component 4	12.175	• System conversion method not well planned: P13 • Inappropriate development methodology used: D2 • Unsuitable working model and prototype: D3 • Programming language and CASE tool selected not adequate: D4 • Difficulty in defining the input and output of system: D7 • Failures and inconsistencies of unit/modules test results: D10 • Failure of user acceptance test: D11 • Time consuming for testing: D12	

A MANOVA analysis of risk components and project performance carried out by Han & Huang (2007) revealed that the composite impact of the planning and systems requirements risk dimensions showed a higher impact on the cost performance of the project. Na *et al.* (2007) reported that functional systems requirements risks were positively correlated with the cost overrun of software projects.

One potentially significant difference between the cost overrun analysis of this section and of the risk occurrence analysis of the previous section is that risk factors comprising this cluster are identified in a single component in the cost overrun analysis whereas they were identified as two separate single risk factor components earlier on. This is to indicate some evidence to suggest that the cost overrun view of risk provides arguable a stronger message in terms of interpretability of risk as it relates to IT software projects than dealing with risk occurrence more generally. This point is elaborated below following further evidence relating to the remaining clusters.

Cluster 2 comprises components 1, 6, and 7 and accounts for over 30% of the variance. This cluster is mainly composed of project development risks all of which have very high loadings and which are positively correlated. In common with the earlier analysis of risk occurrence, a partition of risk factors was observed between those that are unique to the stage in the project cycle which the cluster is mostly closely associated – that is, post-feasibility study – and of risk factors emerging as a consequence of risk crystallisation from inadequate planning and foresight at the feasibility stage or in Cluster 1. Thus, it can be seen that the same risk factors emerging but with some re-organisation into fewer components which taken to be indicative of a stronger and clearer message concerning risk identification and impact. For example, there are now fewer components to interpret as a single cluster compared with the components relating to risk impact.

In Cluster 2, the reduction of components is from 5 to 3 as the research move from risk occurrence to cost overrun. In relation to risk occurrence, components 1 and 3 were separately identified components within Cluster 2. Both of them contain risk factors relating to user involvement. In Cluster 2 of the cost overrun analysis these two components largely merged into a single component. As with the merging of components identified in relation to the feasibility study in the risk occurrence results, there is a similar effect for cost overruns. Again, it appears that taking a cost overrun view of risk perhaps clarifies the commonalities underlying risk that are determined by cost impact. The research argued previously that it was possible to justify the separation of components observed in relation to risk occurrence on the grounds of updating of risk and that, once a project team had become involved, it was possible to discern aspects of risk that would emerge following implementation of some aspect of the project which was a view not available for those involved in the feasibility study specifically. From a cost overrun basis, it does not appear to matter that this division or sequencing of risk factors is relevant. This is observe happening with other components. Thus, component 6 of the cost overrun Cluster 2 has much in common with components 5 and 6 of the risk occurrence analysis. Component 7, the final component of Cluster 2 for cost overrun, appears to map fairly directly with its counterpart component 8 of the risk analysis section.

Cluster 3 is comprised of components 2, 3 and 4 and accounts for 46.01% and in a fairly equal manner. The fact that there were only 2 components identified for the risk occurrence analysis but 3 for the cost overrun appears to contradict the research arguments concerning the consolidating effects of a cost overrun view of risk. Whilst each component for both sections is comprised of risk factors satisfying the extraction cut-off tests, it should be noted that the Cluster 3 of both risk occurrence and cost overrun indicates that risk factors relating to a too narrow focus on technical issues, choice of inappropriate technology, high complexity, immature technology, and out of date technology are key drivers of risk as they relate to technology requirements. Thus, in terms of both risk occurrence and cost overrun there does not appear to be a divergence of opinion in terms of the general thrust of which factors are loading on the respective components.

7.6. Summary

Based on the results from Table 7.3 and Table 7.6, both groups of clusters have total percentage of variance of just under 90 % for the likelihood of occurrence and impact of risk factors on cost overrun. In fact, through the factor analysis process, only less than 13% of information is compromised.

Table 7.11 : Percentage of variance for clusters

Cluster	% variance of likelihood occurrence	% variance of impact of risk factors on cost overrun
Feasibility study	2.56	9.49 %
Project team management	51.52	32.42 %
Technology requirement	35.15	46.01 %

A number of themes have been identified in the survey and analysis of the likelihood of risk factors and of the impact of risk on cost overruns for IT projects. The fact that these were survey responses from IT professionals offers a previously unreported perspective which might be of some value to firms involved with IT projects. The research has identified 3 main clusters of risk for both the likelihood occurrence views and its impact on cost overrun perspective. The 3 main clusters and their associated variances were shown in Table 7.9 and Table 7.10

The research have set out to explore what IT professionals thought of IT software development risks and took the project life cycle as the research risk construct. The research contribution is to offer a different perspective on risk and also to offer an analysis of risk that considers both identification and impact problems separately. There were significant clusterings of risk that suggest the approach adopted is a meaningful one in terms of IT software development for both risk occurrence and risk impact on cost overrun. The sample base and level of explanation offered in terms of variance accounted-for underpin the statistical validity.

A few risk factors may have high occurrence and high impact on cost overrun; or high occurrence but lower impact; or even vice versa. The check and balance of this is needed to assist the companies or practitioners to prioritize them and to develop an appropriate risk mitigation plan accordingly, as whether to reduce the probability of occurrence or focus on the impact of the risk factors on the cost overrun of a software project.

CHAPTER 8 RISK MANAGEMENT STRATEGIES OF SOFTWARE DEVELOPMENT PROJECT

8.1. Introduction

Software projects are especially subject to bounded rationality, induced by cost and schedule constraints, rescue limitations and organizational and technological uncertainty (Bannerman, 2008) (Li et al, 2008). The inconsistencies of performance in software projects with some success as well as failures were still reported in literature even though many research advances had already undertaken. Software project risks can be define as a set of factors or conditions that can pose a serious threat to the successful completion of a software project (Conrow & Shishido, 1997; Huang & Han, 2008; Wallace et al, 2004). Software risk (an uncertain event or condition with negative consequences on a software project) can increase the failure rate of a project if it is ignored.

Evidence indicates that risks in IT projects are not effectively managed and as a result of their lack of identification and management during a project's lifecycle can contribute to their failure. (Willcocks & Griffiths 1997; Hedelin & Allwood 2002; Baccarini et al, 2004). In software projects, the loss may involve increased costs, longer completion times, reduced scope, reduced quality, reduced realization of proposed benefits, or reduced stakeholder satisfaction. In software projects, the monetary cost of poor performance and failure is high but the value of missed benefits is substantial (Bannerman, 2008). Risk management literatures argued that identifying and analysing threats to success allows action to be taken to reduce the chance of failure. Articles have also stressed the importance of empirically categorising the sources and types of risks associated with software development projects (Bannerman 2008; Simister, 2004; Wallace et al, 2004)

The purpose of this chapter is to investigate the effectiveness of risk mitigation strategies in software development projects from the perceptions of the software practitioners. A better understanding of the dimensions of risk mitigation strategies among the software practitioners might assist software project stakeholders to target specific strategies to manage the impact of risk on project areas that are likely to be of high risk.

At this stage, the research look for opinion on risks to IT projects without specifying their relationship to risk likelihood or the impact of cost overrun. That exercise is undertaken in the later chapter as part of the correlation analysis of factor scores in extracted components from risk likelihood and the impact of cost overrun to these mitigation strategies.

Software risk management have been promoted as one approach to reduce project failure and improve software project outcomes (Bannerman, 2008; Boehm, 1991; Keil et al, 2008; Ropponen & Lyytinnen, 2000; Tuman 1993; Remenyi 1999;). Software project risk management is usually defined as a set of principles and practices aimed at identifying, analysing and handling risk factors to improve the chances of achieving a successful project outcome or avoid project failure (Boehm 1989, 1991; Charette 1989; Huang & Han, 2008; Kerzner 2003). According to the Software Engineering Institute (SEI), risk management is a software engineering practice with the processes of assessing continuously what can go wrong, determination of the importance of the risks, and strategies to deal with those risks. Even the Project Management Institute (PMI, 2000) identified the risk management processes as consists of risk identification, risk quantification, risk response and control.

8.2. Software risk management strategies

The whole purpose of software risk management is to identify problems so that action can be taken to eliminate or mitigate their impact. Software risk management usually consists of quantifying the importance of a risk (assessing its probability of occurring and its impact on the project performance) and developing strategies to control it. It is often argued that many of the software project threats have such a low probability, that spending time and money is not justifiable. However, examples and cases in the literature suggest that there is a need to improve the management of threats to software projects (Bannerman, 2008; Brandon, 2005; Conrow & Shishido, 1997; Fairley, 1994; Heemstra & Kusters, 1996). They are many risk analysis techniques currently in use that attempt to evaluate and estimate risk. These techniques can be either qualitative or quantitative depending on the information available and the level of detail that is required. Quantitative and qualitative techniques have their own advantages and disadvantages.

Quantitative techniques rely heavily on statistical approaches which include Monte Carlo Simulation, Fault and Event Tree Analysis; Sensitivity Analysis; Annual Loss Expectancy; Risk exposure; Failure Node and Effect analysis; etc. (Bennet et al, 1996; Ngain & Wat, 2005; Rainer et al 1991). More generic approach for software risk management includes risk lists; risk action lists; risk strategy models and risk strategy analysis (Costa et al 2007; Iversen et al 2004; Jiang et al 2001; Na et al 2007; Ngai & Wat 2005; Keil et al, 2008). Many practice-based approaches also exist. (Prince2, CMMI, COBIT, ITIL, ITGI, NIST, COSO) (Bannerman, 2008)

Iterative risk management steps usually include risk identification, risk analysis, risk response amd risk monitoring and control (Bannerman, IEEE 2008; Ibbs & Kwak, 2000; Simister, 2004). Other approaches to risk management in the research literature, for example (Bannerman, IEEE 2008) :-

- emphasize early development lifecycle risk avoidance in favour of late lifecycle testing to eliminate software defects. (Adler et al, 1999)
- scenario based risk management (Barros et al, 2004)
- Modelling operational risks via Bayesian Networks (Fenton et al, 2002)

- Software risks within a socio technical model of organizational change. (Ropponen & Lyytinnen, 1998; 2000)
- Life cycle based enterprise security risk management. (Drew, 2005)
- Real options approach to managing incomplete knowledge in projects. (Pender, 2001)

Researchers on risk management argue that identifying and analysing threats to success allows actions to be taken to reduce the chance of failure (Wallace et al 2004). Risk management strategies are typically employed in the development process to reduce the risk inherent in software projects. (Boehm,1991; Chittister & Haimes,1994; Fairley,1994; Na et al 2007).

As risks vary in nature, severity and consequences, project managers need to recognize that different types of risks may require different types of strategy or more than one strategy, and a particular kind of strategy may only reduce certain aspect of software development risk but not others (Jiang et al 2001; Cervone 2006). It might be more cost effective to spend resources on preventing problems prior to project development than to wait for problems to appear during the system development.

IT project managers also need to be aware that only very few IT risks have to do with technical issues, and project management is the key strategy for managing risks (Jiang and Klein 2000; (Baccarini et al 2004). Understanding the critical role of project management as a key and encompassing strategy for managing IT project risks is a necessity for project success. The propensity and tendency of IT project managers to become immersed and pre-occupied in technical aspects of their projects, mean that the effective management of IT risks can be impeded.

It is important those risk that are considered to be high level risks, are identified, understood and managed (Baccarini et al 2004) Cervone (2006). This is important to the project success because it allows all team members to identify what the top risks are at any given moment within the project lifecycle. The focusing on the more important risk aspects will allow for more effective management of the project and a narrowing of techniques to mitigate the significant risks (Nidumolu 1996; Jiang and Klein 2000).

8.3. Risk management strategies literatures

The PMBOK describes risk response planning as part of a systematic process of risk management. It is a development of options and actions to threats, either in the form of strategies to avoid risks or to mitigate the impact if it occurs. Risk avoidance techniques are meant to either eliminate the risk or protect the project from the impact of the risk. Alternatively, risk mitigation techniques intend to reduce the impact of an existing risk to an acceptable threshold.

As the study of risk factors occurrence and its impact on cost overrun of software project is based on the project management framework perspective, this part of study are focused on the more generic risk management strategies as described in Table 8.1 below. The specific strategies of qualitative or quantitative risk management techniques mentioned in the previous paragraph will not be analysed further, as only the generic strategies that is more relevant to main list of risk factors is chosen. These generic risk management strategies was also chosen without having to go into greater detail about the specific type of project or specific business environment, so that, the strategies could possibly be applied to any kind of software project or business environment. However, not many researchers had discussed in detail the risk mitigation strategies of a software project.

Table 8.1 : Research undertaken in IT risk management

YEAR	RESEARCHER	RESEARCH AREA	Risk mitigation strategies	Point of view
2009	Mahaney & Lederer	Role of monitoring and shirking in IS project management	Monitoring of project Regular updating of project against goals	Project managers
2008	Shih-Chieh Su et al	Impact of user review on software responsiveness	Users involvement	Project managers
2007	Dey et al	Risk management framework for software development projects from developer's perspective, using a case study of public sector organization in Barbados.	User's involvement Scope management planning Establish clear client requirements Resource planning Process reengineering Benchmarking Effective communications Unit or independence testing Establish scope Develop work breakdown structure Control mechanism	Developers
2007	Tesch et al	IT project risk perspective of project management professionals (PMP)	Team communication Project managers leading role and experience User customer support Top management backing Plan project in phases Project planning Proper budgeting Develop resource allocation planning Contingency plan to maintain project Re-evaluate project CBA Use change management process Conduct feasibility study Pilot and prototype technology before rolling into organization Alternative technology and development methodology Clear scope requirements Project control mechanism User participation commitment Develop approach to get feedback Set up key milestones	Views of project management professionals (PMP)
2004	Wallace & Keil	Identification of risks that posed threat to successful project outcomes	Strategies related to project scope and requirements Strategies related to project execution Experienced project team members Experienced project managers Project planning and control techniques Identified scope and requirements	Project managers

2004	Wallace et al	Investigation of dimensions of risk and an exploratory model, on the software project performance.	Planning control mechanism Assembling high skilled project team Training User involvement Top management involvement Counter risk associated with organizational environment, users, requirements, project complexity. Good project management practices	Project managers
2004	Baccarini et al	In-depth interviews with IT professionals from leading firms in Western Australia to determine how IT risks were managed in their projects, where the respondents ranked IT risks in terms of likelihood and consequences to identify the most important risks.	Manage the relationship Project planning and schedule management Manage expectations Obtain management support Develop customer relationship Maintain market entry barrier Establish sound requirements Plan for resources Plan contingency options Assess project staff capability Change project management objectives Manage stakeholders Executive management support Clear scope definition Develop clear requirements definition Adequate documentation Perform group reviews Progressive signoff of milestones Comprehensive testing Users supports Formsl change management process Consult/educate users Monitoring project Project managers experience Roles and responsibilities clearly defined Clear coomunication External consultants	Interview with IT professionals
2000	Cule et al	Strategies for heading off IS project failures	Top management commitment Users involvement Effective communication within project team Regular updating Monitoring of projects Assessment mechanism Benchmarking with other projects Lessons learned from previous projects Task management approach	IS project managers
2000	James Jiang & Gary Klien	Impact of the spectrum of risks on different aspect of systems development and project effectiveness	Interpersonal and team skills Skills training User participation and user commitment Clearly defined roles Clear project scope and task Clear communications User experience Control of conflicts	Survey of project managers

Study by Tesch et al (2007) identified many different strategies for avoiding and mitigating the impact of the risks in software project. There were also similar strategies suggested for more than one risk. The strategies suggested includes consistent commitment of top management, planning and scheduling of project in phases, users involvement, good communication lines and resource requirement planning. The study also suggested the project team to have a written project charter which contains information such as a clear goals and requirements, clearly outlined deliverables and success criteria, contingency or back up plan and roles and responsibilities matrix.

Shih-Chieh Su et al (2008) also highlighted on the users perspective to act to control progress and act as product quality gate keepers in the software development process. The study mentioned that users should not only play the role as requirements providers but should continue to engage in the subsequent development process to make sure that user requirements were fulfilled. Other researchers added the important of active participations from the users includes Cule et al (2000), Jiang and Klien (2000), Baccarini et al (2004), Wallace et al (2004) and Dey et al (2007). Most of the studies revealed through user participation influences the final system outcomes by mitigating the effect of various risk factors, reduce the risk uncertainties and improve the product during the software development process.

Mahaney et al (2009) focus the impact of monitoring and regular updating of project as important strategies. The study revealed that through monitoring and regular updating, could provide information that project is progressing within budget, schedule and quality expectations. With this information, subjective assessment can be made whether the overall benefits of the software project can be realized. The research showed that increased monitoring can reduce the project failure by reducing over-commitment and encourages subordinates to act in the interests of their managers. The research suggest that monitoring and regular tracking of project progress can inspires developers to refrain from loafing and poor focus. Monitoring can also be done through project management softwares such as Microsoft Project, Primavera, Gantt Chart or Critical path analysis. Cule et al (2000), Baccarini et al (2004) and McChesney & Gallagher (204) also added the important of monitoring and regular updating, as very important strategies.

Other generic risk management strategies includes top management commitment, effective communications, project managers' experience, clear requirements and project scope (Cule et al, 2000; Jiang et al, 2000; Wallace et al, 2004, Baccarini et al, 2004; Dey at al, 2007); quality control mechanism (Dey et al, 2007; Osmundson et al, 2003); testing (Frankl & Weyuker, 2000; Baccarini et al, 2004); training (Wallace et al, 2004; Jiang & Klien, 2000; Subramanian et al, 2007); and also prototyping (Subramanian, 2007; Tesch et al, 2007). Even, studies of project management success criteria on software project by Proccacino et al (2005), Pereira et la (2007), Verner et al (1999), Palitha et al (2002), Milis (2002), Pinto J.K. (1998), and Tan R.R (1996) also focused on the strategies like users involvement, top management commitment, good project management practices, clear requirements, project planning and scheduling, and clear goals objectives.

Most of the studies by researchers on managing risk in software development project or IT projects, identify and priorities risks through empirical research in order to suggest mitigating measures. The works were also based on anecdotal evidence and on studies limited to a narrow portion of the development processes, or even on the broad perspectives of general project performance. The framework, guidelines or systematic models of risk management proposed by these previous researches, predominantly deal with specific techniques and were more focused on the project managers perspectives. But, the generic risk management strategies had not being studied to include the perspectives of the software development team personnel of Project managers, Developer, the IT support staff and also the management hierarchy themselves. This chapter will highlighted the effectiveness of the generic risk management strategies.

8.4. Method

The main methodology for the research has been explained in previous Chapter 2 (Research methodology). This chapter will outlined the method used for the risk management strategies questionnaires in the research.

The number of risk strategies was extracted from a review of literatures of risk management associated with software development projects. The list of risk strategies that has been established are shown in Table 8.1. The extracted risk strategies were also validated in a pilot study with experienced academics. The purpose for conducting such a pilot study is to test the potential response rate, suitability, and comprehensibility of the questionnaire, and also to review the design and structure of the survey.

A number of the strategies were very specific techniques and modelling which requires some technical knowledge and skills, and some were quite generic. However, since the new framework proposed for the research was based on the project management perspectives, only the generic strategies were chosen with the additional of strategies that were also relevant to the risk factors in the main part of the questionnaire. The main purpose of this set of generic strategies is to get the idea and perceptions of the practitioners of the risk management exercise which do not have to be very technical related issues, and could easily be followed by other non-IT related staff within the organization. In a way, these generic strategies may also be used for most software project, without having to worry about the specific type of project and business environment.

Based on the findings from the pilot study, some strategies were omitted and others were added to the list. A final questionnaire survey was designed to elicit data about the effectiveness of risk strategies. In the main survey, 30 strategies were included in the questionnaires. Respondents were asked to rate the effectiveness of the risk strategies in response to the risk factors using the Likert scale of 0-6 as below :-

a. 0 – don't know
b. 1 – not effective strategy
c. 2 - very slightly effective strategy
d. 3 – generally effective strategy
e. 4 – highly effective strategy
f. 5 – very highly effective strategy
g. 6 – exceptionally effective strategy

This is to give a general idea of the importance of certain strategies compared to the other strategies, without going into greater detail of the interpretation and magnitude value of the effectiveness.

Table 8.2 : Risk mitigation strategies used in the research

No	Ref	RISK MITIGATION STRATEGIES
1	S1	Define a clear goals and objectives of the project
2	S2	Conduct a through analysis feasibility study
3	S3	Use of project tracking system and regular updating
4	S4	Proper project planning and scheduling
5	S5	Identify critical and non-critical activities
6	S6	Set key performance indicators and standards for stages/processes
7	S7	Lesson learned from past software development projects
8	S8	Identify success criteria
9	S9	Consistent commitment of management
10	S10	Quality control procedure
11	S11	Risk management methodology/techniques/tools
12	S12	Hire external expertise/consultant
13	S13	Contingency plan
14	S14	Conduct pilot testing
15	S15	Prototyping
16	S16	Thorough analysis of development methodology
17	S17	Proper timeframe for testing
18	S18	Conduct a thorough user acceptance test
19	S19	Planned for parallel or phased conversion
20	S20	Developed a clear and detail requirements
21	S21	Incorporate alternative development methodology
22	S22	Backup the system thoroughly
23	S23	Software security checklist and authentication process
24	S24	Cost control procedure
25	S25	Technical support team
26	S26	Proper planning of resources
27	S27	Effective training for staff
28	S28	Effective lines of communication
29	S29	Good project management and leadership
30	S30	Greater degree of users involvement and and commitment

The statistical methods of mean, standard deviation and Kendall's test were used to rank the effectiveness or risk strategies. One-way analysis of variance (ANOVA) was used to compare the means of respondents and to determine if there were any significant differences among them. The Kendall (W) test was undertaken to determine whether there were differences between respondents' rankings of risk mitigating strategies. The research have used Tukey-B test to check the validity of our results. This was necessary because of the uneven sample sizes of the IT professional respondents. To interpret the results it was important to look at the overall Fs values and chi-square, degree of freedom. These values are used to indicate whether there is a difference between respondents (p values < 0.05).

8.5. Respondent's information

As had been explained in previous Chapters, more than half of the respondents (61%) had more than 10 years of experience in software project development with an overall average of 11.8 years and standard deviation of 5.29. Furthermore the overall average numbers of software projects that respondents were involved with or had undertaken were 9 with a standard deviation of 5.31. It can be argued that the respondents have good working knowledge and insight into software development projects and processes. The wealth of experience among the respondents was relevant and significant and provides a good experience base to support questionnaire responses. This may give reasonable support for the concluding arguments in the study.

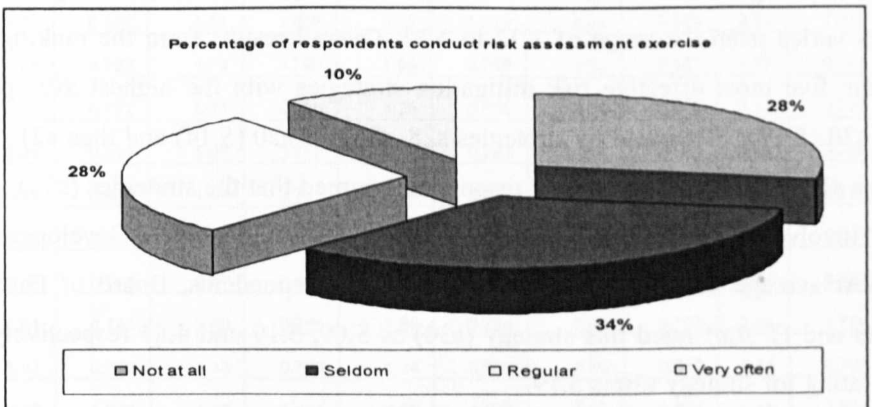


Figure 8.1 : Percentage of respondents conduct risk assessment exercise

The respondents were also asked whether risk assessment exercise was carried out for their software project. From the responses, 28% of the respondents did not at all conduct risk assessment exercise, while the other 72% respondents conducted the risk assessment exercise for their software project occasionally and regularly. Furthermore, 68% of the respondents consider their companies of having between 1-4 expert professionals of risk management. However, 40% of the respondents were not satisfied with their risk management practice of software project.

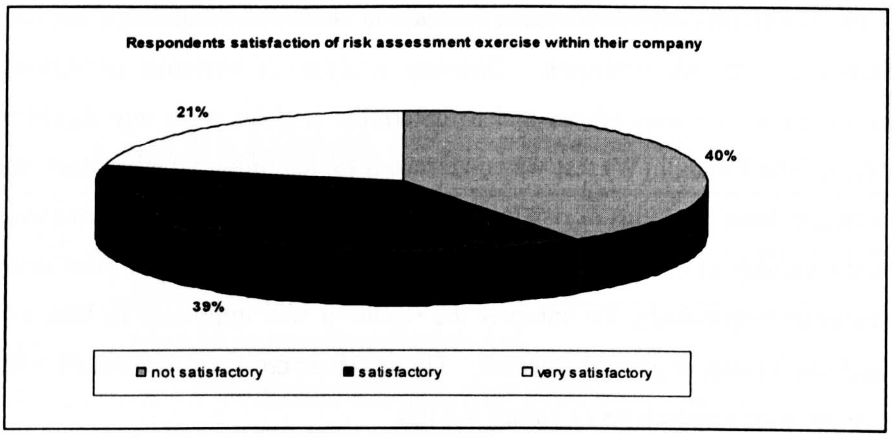


Figure 8.2 : Respondents' satisfaction of risk assessment exercise

The purpose of the ranking is to extract the most and least important risk mitigation strategies based the perception of the survey participants. Responses to the rating of risk mitigation strategies were on a scale numbered 0–6, from don't know to exceptionally effective strategy. Scores from these answers enabled a mean and standard deviation scores to be derived for the 30 risk mitigation strategies. These measures are also computed for each type of survey respondent. These measures are then used to rank order risk mitigation strategies according the perception of the survey respondents' four categories.

8.6. Average rating

The rating of the strategies is shown in Table 8.3. The average rating for the mitigation strategies varies from the range of 1.32 to 5.31. Overall results from the ranking revealed that among the five most effective risk mitigation strategies with the highest average score is the strategy s30 (5.19), followed by strategies s28, s26 and s20 (5.14) and then s21 (5.08). Lowest of all was s19 (1.77). All 4 groups of respondents agreed that the strategies (s30), greater degree of users' involvement and commitment, came top of the ranking, where Developer rated s30 with the highest average of 5.31. The other 3 groups of respondents, Board of Directors, Project Managers and IT staff rated this strategy (s30) as 5.09, 5.19 and 4.84 respectively. The overall average rating for strategy s30 is 5.19.

A close scrutiny of the results reveals that the two most effective strategies (s30, s28) as perceived by the board of directors are mostly consistent with those perceived by project managers and developers. However, IT support staff perceived risk strategy (s28) to be less important than risk strategy (s29). The ranking also revealed that strategy (s29) is perceived the 2nd important by directors while project managers and developers perceived otherwise. Also strategy (s29) was perceived by IT staff as less important. Such difference in ranking the order of risk strategies effectiveness is an indicative of potential differences in roles and responsibilities in the management of software projects.

Table 8.3 : Average rating for the risk mitigation strategies

Strategy	Board		Project Manager		Developer		IT staff		Overall		Kendall mean rank
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	
S1	5.02	0.802	5.01	0.718	5.14	0.707	4.76	0.723	5.04	0.731	22.85
S2	4.33	1.012	4.50	1.085	4.61	1.078	4.12	0.927	4.49	1.066	18.78
S3	2.78	0.786	2.79	0.823	2.88	0.818	2.48	0.770	2.80	0.815	8.53
S4	4.98	0.774	5.04	0.711	5.11	0.714	4.84	0.688	5.04	0.720	22.77
S5	5.02	0.802	5.01	0.718	5.14	0.707	4.76	0.723	5.04	0.731	22.85
S6	2.61	0.714	2.75	0.817	2.81	0.816	2.40	0.645	2.73	0.796	8.61
S7	5.02	0.802	5.00	0.723	5.14	0.707	4.76	0.723	5.04	0.733	22.83
S8	5.02	0.745	5.05	0.705	5.11	0.714	4.84	0.688	5.05	0.713	22.85
S9	3.39	1.085	3.33	1.037	3.63	1.052	2.92	0.954	3.42	1.057	11.34
S10	1.78	0.841	1.82	0.809	1.86	0.794	1.44	0.651	1.80	0.801	3.90
S11	4.37	1.123	4.42	1.089	4.68	1.069	3.92	0.954	4.47	1.091	18.50
S12	1.67	0.701	1.81	0.793	1.89	0.814	1.40	0.645	1.79	0.786	3.78
S13	4.22	1.114	4.13	1.025	4.42	1.089	3.76	0.831	4.22	1.061	16.83
S14	4.37	1.040	4.65	1.074	4.77	1.041	4.16	1.028	4.62	1.066	19.56
S15	4.98	0.830	4.93	0.714	5.04	0.744	4.72	0.678	4.96	0.741	22.23
S16	1.72	0.750	1.90	0.791	1.99	0.779	1.48	0.714	1.88	0.785	4.09
S17	2.70	0.785	2.61	0.773	2.77	0.810	2.24	0.523	2.65	0.782	7.90
S18	3.33	0.967	3.51	1.006	3.72	1.029	3.08	0.909	3.53	1.015	12.25
S19	1.78	0.728	1.79	0.783	1.85	0.769	1.32	0.557	1.77	0.764	3.61
S20	5.07	0.772	5.11	0.730	5.26	0.659	4.80	0.645	5.14	0.713	23.53
S21	5.04	0.893	5.07	0.755	5.13	0.723	4.80	0.764	5.06	0.766	23.01
S22	1.80	0.806	1.87	0.771	2.06	0.731	1.32	0.557	1.89	0.769	4.17
S23	3.30	1.030	3.36	1.033	3.44	1.067	3.00	1.041	3.35	1.047	11.32
S24	2.74	0.773	2.77	0.801	2.98	0.784	2.32	0.557	2.81	0.791	8.82
S25	2.61	0.745	2.60	0.765	2.68	0.794	2.40	0.707	2.61	0.769	7.76
S26	5.07	0.772	5.13	0.706	5.26	0.659	4.80	0.645	5.14	0.703	23.58
S27	4.39	1.000	4.42	1.011	4.50	1.036	4.00	1.041	4.41	1.024	18.12
S28	5.07	0.772	5.13	0.706	5.26	0.659	4.80	0.645	5.14	0.703	23.58
S29	5.09	0.755	5.08	0.702	5.14	0.719	4.76	0.779	5.08	0.725	23.14
S30	5.09	0.812	5.19	0.714	5.31	0.636	4.84	0.624	5.19	0.704	23.93

It is interestingly to observe from the list of 30 risk strategies, all 4 groups of respondents rated the top 10 strategies with 4.76 and more. The top 10 strategies were dominated by S30, S29, S28, S26, S21, S20, S8, S5, S4 and S1. Strategies such as clear objectives and requirements, planning of scheduling ad resources, identification of success criteria and critical activities, project leadership, users' commitment and effective lines of communication were deemed important strategies. It is important to point out that these top 10 rated mitigation strategies, predominantly were non-technical risks, or some literatures might categorized as project planning or organizational issues or project management related matters.

The respondents commonly agreed that the top 10 mitigation strategies mentioned above would have a highly and very highly effect on the effectiveness of the risk mitigation strategies in relation of the risk reduced.

From Table 8.3, it can also be seen that strategies S10, S12 and S19, were consistently rated as the least effective in reducing then risks. This includes strategies such as quality control procedure, hiring of external consultant or expertise and undertaking a parallel or phased conversion. These strategies have an average rating of below 2.0, which mean not an effective strategy.

It is interesting to see that quality control procedures are less effective than clear requirements and effective lines of communication risk reduction strategies. This finding correspondent with a view that the survey participant might perceive that quality control is part of project management function.

The four categories of respondent mean scores are shown in Table 8.3. Clearly the figure illustrates that the effectiveness of risk mitigating strategy are classified into four main results. First, the most effective strategies as sighted by all respondents are (S30, S29, S28, S26, S21, S20, S15, S8, S7, S5, S4, S1). Secondly, the least effective strategies are (S22, S16, S12, S10, and S19). Thirdly, the figure noticeably shows that the developers consistently rated all the mitigation strategies higher than the other respondents whereas IT support staff perceived them less effective. And the fourth main results, ironically, the managing directors and project managers perception is nearly the same for all the strategies except for the strategy S13.

To support the argument for Table 8.3 above, SPSS software was used to carry out Kendall's coefficient of concordance test (W), to measure the agreement among the raters or respondents. Kendall (W) ranges between 0 (non-agreement) and 1 (complete agreement). From the Kendall (W) rank test, the mean rank of the risk factors were in the range of as low as 3.61 and as high as 23.93. The high mean rank being the factors such as S30, S29, S28, S26, S21, S20, S8, S5, S4 and S1. This is consistent with the rating by the respondents. With significant level of 0.05, and the null hypothesis ($H_0 > 0.05$) as being no significant agreement among the respondents, the results in Table 8.4 from the Kendall's rank test showed the W-value of 0.842. This test tells that there is a significant agreement among the respondents.

This means that, there are common agreements, consensus or concordance among the respondents for the main overall results in terms of, what are the important strategies and the least important strategies. This consensus and consistency is important for the inference of these results for the general populations. This Kendal coefficient concordance test only proved the general consensus of the main findings of the analysis. However, this does not mean that all the respondents gave the same rating for all strategies. This will be explained and tested later in Section 8.7 using the ANOVA test.

Table 8.4: Kendall's Coefficient of Concordance

N	324
Kendall's W	0.842
Chi-Square	7910.714
df	29
Asymp. Sig.	0.000

8.7. Anova analysis

The statistic measures reported in Table 8.3, indicate that the values of the means, standard deviations and variations, of the four groups (Board, Project manager, Developer, IT staff) of respondents were comparatively close. These results suggest there seems to be little or no significant differences between the groups' responses. Hence, ANOVA analysis was conducted in order to detect and justify the groups' responses statistical differences. The testing hypothesis was:

H_0 ($p > 0.05$): There is no significant difference among the respondents rating for the effectiveness of the risk mitigation strategies.

H_1 ($p < 0.05$) : At least one of the groups rating for the effectiveness of the risk mitigation strategies significantly different from at least one other groups.

The output of the ANOVA analysis of each risk factors is shown in Table 8.5.

The ANOVA in Table 8.5 shows whether the overall *Fs* values for these risk mitigation strategies were significant. A statistically significant difference was found among some of the risk mitigation strategies for the respondents. The 4 group of respondents' responses differ significantly on 15 of the 30 mitigation strategies which include S9, S11, S12, S13, S14, S16, S17, S18, S19, S20, S22, S24, S26, S28 and S30. This means that at least one of the groups rating for the risk mitigation strategies significantly different from at least one other group. In this case, reject null hypothesis.

However, ANOVA analysis only tells whether there is have sufficient evidence to state that the rating for the strategies by one group differ significantly from other. It will not tell which specific means are different from which other ones. In order to know this, a follow up test called Post Hoc Multiple Comparison Test was conducted. Due to the uneven sample sizes, a Tukey test was done. However, only the 15 risk strategies that were significantly different were shown in Figure 8.3.

Table 8.5 : The F-value and Significant value of the ANOVA analysis.

		Sum of Squares	df	Mean Square	F	Sig.
S1	Between Groups	3.397	3	1.132	2.143	0.095
	Within Groups	169.082	320	0.528		
	Total	172.478	323			
S2	Between Groups	6.386	3	2.129	1.889	0.131
	Within Groups	360.565	320	1.127		
	Total	366.951	323			
S3	Between Groups	3.380	3	1.127	1.707	0.165
	Within Groups	211.175	320	0.660		
	Total	214.556	323			
S4	Between Groups	1.756	3	0.585	1.131	0.337
	Within Groups	165.639	320	0.518		
	Total	167.395	323			
S5	Between Groups	3.397	3	1.132	2.143	0.095
	Within Groups	169.082	320	0.528		
	Total	172.478	323			
S6	Between Groups	4.261	3	1.420	2.269	0.080
	Within Groups	200.292	320	0.626		
	Total	204.552	323			
S7	Between Groups	3.466	3	1.155	2.174	0.091
	Within Groups	170.089	320	0.532		
	Total	173.556	323			
S8	Between Groups	1.565	3	0.522	1.027	0.381
	Within Groups	162.543	320	0.508		
	Total	164.108	323			
S9	Between Groups	12.360	3	4.120	3.784	***0.011***
	Within Groups	348.390	320	1.089		
	Total	360.750	323			
S10	Between Groups	3.808	3	1.269	1.996	0.115
	Within Groups	203.550	320	0.636		
	Total	207.358	323			
S11	Between Groups	13.438	3	4.479	3.861	***0.010***
	Within Groups	371.253	320	1.160		
	Total	384.691	323			
S12	Between Groups	5.682	3	1.894	3.123	***0.026***
	Within Groups	194.047	320	0.606		
	Total	199.728	323			
S13	Between Groups	11.382	3	3.794	3.449	***0.017***
	Within Groups	352.059	320	1.100		
	Total	363.441	323			

S14	Between Groups	11.007	3	3.669	3.302	* ***0.021***
	Within Groups	355.536	320	1.111		
	Total	366.543	323			
S15	Between Groups	2.349	3	0.783	1.430	0.234
	Within Groups	175.206	320	0.548		
	Total	177.556	323			
S16	Between Groups	6.756	3	2.252	3.747	* ***0.011***
	Within Groups	192.306	320	0.601		
	Total	199.062	323			
S17	Between Groups	6.192	3	2.064	3.457	* ***0.017***
	Within Groups	191.092	320	.597		
	Total	197.284	323			
S18	Between Groups	11.297	3	3.766	3.749	* ***0.011***
	Within Groups	321.453	320	1.005		
	Total	332.750	323			
S19	Between Groups	5.840	3	1.947	3.409	* ***0.018***
	Within Groups	182.713	320	0.571		
	Total	188.552	323			
S20	Between Groups	5.031	3	1.677	3.375	* ***0.019***
	Within Groups	158.994	320	0.497		
	Total	164.025	323			
S21	Between Groups	2.233	3	0.744	1.271	0.284
	Within Groups	187.406	320	0.586		
	Total	189.639	323			
S22	Between Groups	11.911	3	3.970	7.103	* ***0.000***
	Within Groups	178.864	320	0.559		
	Total	190.775	323			
S23	Between Groups	4.132	3	1.377	1.260	0.288
	Within Groups	349.757	320	1.093		
	Total	353.889	323			
S24	Between Groups	9.979	3	3.326	5.539	* ***0.001***
	Within Groups	192.157	320	0.600		
	Total	202.136	323			
S25	Between Groups	1.655	3	0.552	0.934	0.425
	Within Groups	189.119	320	0.591		
	Total	190.775	323			
S26	Between Groups	4.950	3	1.650	3.417	* ***0.018***
	Within Groups	154.520	320	0.483		
	Total	159.469	323			
S27	Between Groups	5.190	3	1.730	1.661	0.175
	Within Groups	333.390	320	1.042		
	Total	338.580	323			
S28	Between Groups	4.950	3	1.650	3.417	* ***0.018***
	Within Groups	154.520	320	.483		
	Total	159.469	323			
S29	Between Groups	3.047	3	1.016	1.948	0.122
	Within Groups	166.867	320	0.521		
	Total	169.914	323			
S30	Between Groups	5.355	3	1.785	3.690	* ***0.012***
	Within Groups	154.781	320	0.484		
	Total	160.136	323			

From Figure 8.3(Tukey Post Hoc test), with the significant values of less than 0.05 ($p < 0.05$) and the different in the subset, shows that there is significant difference between the respondents' responses. For example, IT staff and Developers have significant rating for strategies S9, S11, S13, S14, S18, S20, S26, S28. IT staff also has different rating from the other 3 practitioners for strategies S17, S19, S22, S24. Managing Directors and Project managers seemed to have a consensus of agreements in the ratings of the strategies.

The ANOVA and Post Hoc test proved that the respondents rated the strategies quite differently even though the mean scores of the strategies were quite close together, whereas, the Kendal concordance test in the previous Section 8.6 shows, there are common consensus of the overall main results.

Furthermore, through ANOVA analysis of the top 10 ranked strategies (S30, S29, S28, S26, S21, S20, S8, S5, S4, S1), only 4 strategies (S30, S28, S26, S20) were found to have a statistically significance difference.

The overall F-values for this 4 strategies were; $F(3,320)=3.690$, $p=0.012$; $F(3,320)=3.417$, $p=0.018$; $F(3,320)=3.417$, $p=0.018$; and $F(3,320)=3.375$, $p=0.019$; respectively. Because of the difference in group sizes, Post Hoc Tukey HSD was utilized in testing to determine which groups differ from each other. The result from test shows that Developer and IT Staff differed significantly in their responses for strategies S30 ($p=0.012$); S28 ($p=0.014$); S26 ($p=0.014$); and S20 ($p=0.016$).This difference in the mean score between these two groups is clearly shown in Table 8.3.

Hence, to confirm that there is significant agreement between the four categories in the survey, the research have used Kendal's nonparametric test. The measure of the relationship between rankings of the effectiveness of risk mitigation strategies for each respondent category is used to investigate the agreement or concordance between the survey respondent in their precipitation on the effectiveness of risk strategies in reduction of the risks. Kendal's coefficient of concordance provides a measure of agreement between category of respondent, and concordance between rankings of risk mitigation strategies. It ranges between "0" and "1", with "0" indicating no agreement and "1" designating perfect concordance. **Table 8.6** portrays the statistical findings of this analysis. It is shown that those values of Kendall's coefficient range between 0.83 and 0.88 for the four categories. These high values of Kendall's coefficient indicate strong agreement between survey respondent on ranking the effectiveness of risk mitigation strategies. The values of significance level are all at $P = 0.00$. These values indicate that, the null hypothesis: there is no agreement between survey respondent, has to be rejected ($p < 0.05$). The alternative hypothesis that, there is a significant agreement between the four categories, is acceptable with confidence limit $p > 95\%$.

S9	N	Subset for alpha = 0.05	
		1	2
IT staff	25	2.92	
project manager	135	3.33	3.33
board of directors	46	3.39	3.39
developer	118		3.63

S11	N	Subset for alpha = 0.05	
		1	2
IT staff	25	3.92	
board of directors	46	4.37	4.37
project manager	135	4.42	4.42
developer	118		4.68

S12	N	Subset for alpha = 0.05	
		1	2
IT staff	25	1.40	
board of directors	46	1.67	1.67
project manager	135		1.81
developer	118		1.89

S13	N	Subset for alpha = 0.05	
		1	2
IT staff	25	3.76	
project manager	135	4.13	4.13
board of directors	46	4.22	4.22
developer	118		4.42

S14	N	Subset for alpha = 0.05	
		1	2
IT staff	25	4.16	
board of directors	46	4.37	4.37
project manager	135	4.65	4.65
developer	118		4.77

S17	N	Subset for alpha = 0.05	
		1	2
IT staff	25	2.24	
project manager	135		2.61
board of directors	46		2.70
developer	118		2.77

S16	N	Subset for alpha = 0.05	
		1	2
IT staff	25	1.48	
board of directors	46	1.72	1.72
project manager	135		1.90
developer	118		1.99

S18	N	Subset for alpha = 0.05	
		1	2
IT staff	25	3.08	
board of directors	46	3.33	3.33
project manager	135	3.51	3.51
developer	118		3.72

S19	N	Subset for alpha = 0.05	
		1	2
IT staff	25	1.32	
board of directors	46		1.78
project manager	135		1.79
developer	118		1.85

S20	N	Subset for alpha = 0.05	
		1	2
IT staff	25	4.80	
board of directors	46	5.07	5.07
project manager	135	5.11	5.11
developer	118		5.26

S22	N	Subset for alpha = 0.05	
		1	2
IT staff	25	1.32	
board of directors	46		1.80
project manager	135		1.87
developer	118		2.06

S24	N	Subset for alpha = 0.05	
		1	2
IT staff	25	2.32	
board of directors	46		2.74
project manager	135		2.77
developer	118		2.98

S26	N	*Subset for alpha = 0.05*	
		1	2
IT staff	25	*4.80*	
board of directors	46	*5.07*	*5.07*
project manager	135	*5.13*	*5.13*
developer	118		*5.26*

S28	N	Subset for alpha = 0.05	
		1	2
IT staff	25	4.80	
board of directors	46	5.07	5.07
project manager	135	5.13	5.13
developer	118		5.26

S30	N	Subset for alpha = 0.05	
		1	2
IT staff	25	4.84	
board of directors	46	5.09	5.09
project manager	135		5.19
developer	118		5.31

- Groups listed in the same subset are not significantly different, but groups in different subset are significantly different. For example, the rating for S9 and S11 are not different for Project manager, Board of Directors and Developers. But, only IT staff and Developer rated the strategy differently.

Figure 8.3 : Tukey HSD Post Hoc Multiple Comparison Test

Table 8.6 : Kendal concordance analysis using SPSS

Respondents' category	Degree of freedom	Chi-square	Kendal's coefficient (W)	Significance
Board	29	1109.789	0.832	0.000
Project Manager	29	3247.840	0.830	0.000
Developer	29	2926.970	0.855	0.000
IT staff	29	643.489	0.888	0.000

Considering the ANOVA analysis and Post Hoc comparison test for the effectiveness of the risk mitigation strategies, it can be said that the difference among the means are significant. It can be accepted that there is a genuinely significant overall difference among the practitioners in their rating of the risk mitigation strategies.

Table 8.7 : The Kendal mean rank and significant value for Anova

	Board			Project Manager			Developer			IT staff			Overall			K	Sig Value
	M	SD	R	M	SD	R	M	SD	R	M	SD	R	M	SD	R		
S1	5.02	0.802	10	5.01	0.718	10	5.14	0.707	8	4.76	0.723	11	5.04	0.731	10	22.85	0.095
S2	4.33	1.012	16	4.50	1.085	14	4.61	1.078	15	4.12	0.927	14	4.49	1.066	14	18.78	0.131
S3	2.78	0.786	21	2.79	0.823	21	2.88	0.818	22	2.48	0.770	21	2.80	0.815	22	8.53	0.165
S4	4.98	0.774	12	5.04	0.711	8	5.11	0.714	11	4.84	0.688	3	5.04	0.720	8	22.77	0.337
S5	5.02	0.802	9	5.01	0.718	9	5.14	0.707	7	4.76	0.723	10	5.04	0.731	9	22.85	0.095
S6	2.61	0.714	25	2.75	0.817	23	2.81	0.816	23	2.40	0.645	22	2.73	0.796	23	8.61	0.080
S7	5.02	0.802	8	5.00	0.723	11	5.14	0.707	6	4.76	0.723	9	5.04	0.733	11	22.83	0.091
S8	5.02	0.745	7	5.05	0.705	7	5.11	0.714	10	4.84	0.688	2	5.05	0.713	7	22.85	0.381
S9	3.39	1.085	18	3.33	1.037	20	3.63	1.052	19	2.92	0.954	20	3.42	1.057	19	11.34	***0.011***
S10	1.78	0.841	27	1.82	0.809	28	1.86	0.794	29	1.44	0.651	27	1.80	0.801	28	3.90	0.115
S11	4.37	1.123	15	4.42	1.089	15	4.68	1.069	14	3.92	0.954	16	4.47	1.091	15	18.50	* ***0.010***
S12	1.67	0.701	30	1.81	0.793	29	1.89	0.814	28	1.40	0.645	28	1.79	0.786	29	3.78	* ***0.026***
S13	4.22	1.114	17	4.13	1.025	17	4.42	1.089	17	3.76	0.831	17	4.22	1.061	17	16.83	* ***0.017***
S14	4.37	1.040	14	4.65	1.074	13	4.77	1.041	13	4.16	1.028	13	4.62	1.065	13	19.56	* ***0.021***
S15	4.98	0.830	11	4.93	0.714	12	5.04	0.744	12	4.72	0.678	12	4.96	0.741	12	22.23	0.234
S16	1.72	0.750	29	1.90	0.791	26	1.99	0.779	27	1.48	0.714	26	1.88	0.785	27	4.09	* ***0.011***
S17	2.70	0.785	23	2.61	0.773	24	2.77	0.810	24	2.24	0.523	25	2.65	0.782	24	7.90	* ***0.017***
S18	3.33	0.967	19	3.51	1.006	18	3.72	1.029	18	3.08	0.909	18	3.53	1.015	18	12.25	* ***0.011***
S19	1.78	0.728	28	1.79	0.783	30	1.85	0.769	30	1.32	0.557	29	1.77	0.764	30	3.61	* ***0.018***
S20	5.07	0.772	5	5.11	0.730	4	5.26	0.659	4	4.80	0.645	6	5.14	0.713	4	23.53	* ***0.019***
S21	5.04	0.893	6	5.07	0.755	6	5.13	0.723	9	4.80	0.764	7	5.06	0.766	6	23.01	0.284
S22	1.80	0.806	26	1.87	0.771	27	2.06	0.731	26	1.32	0.557	30	1.89	0.769	26	4.17	* ***0.000***
S23	3.30	1.030	20	3.36	1.033	19	3.44	1.067	20	3.00	1.041	19	3.35	1.047	20	11.32	0.288
S24	2.74	0.773	22	2.77	0.801	22	2.98	0.784	21	2.32	0.557	24	2.81	0.791	21	8.82	* ***0.001***
S25	2.61	0.745	24	2.60	0.765	25	2.68	0.794	25	2.40	0.707	23	2.61	0.769	25	7.76	0.425
S26	5.07	0.772	4	5.13	0.706	3	5.26	0.659	3	4.80	0.645	5	5.14	0.703	3	23.58	* ***0.018***
S27	4.39	1.000	13	4.42	1.011	16	4.50	1.036	16	4.00	1.041	15	4.41	1.024	16	18.12	0.175
S28	5.07	0.772	3	5.13	0.706	2	5.26	0.659	2	4.80	0.645	4	5.14	0.703	2	23.58	* ***0.018***
S29	5.09	0.755	2	5.08	0.702	5	5.14	0.719	5	4.76	0.779	8	5.08	0.725	5	23.14	0.122
S30	5.09	0.812	1	5.19	0.714	1	5.31	0.636	1	4.84	0.624	1	5.19	0.704	1	23.93	* ***0.012***

- M-mean; SD-standard deviation; R-ranking; K-Kendal mean rank
- The shaded denotes the top 10 ranking of each category of respondents
- The bold asterisk (*) denotes the strategies that was significant from Anova analysis

8.8. Discussion

The ANOVA analysis and the Post Hoc test also revealed that IT staff has significant difference of rating of the risk strategies compared to the other 3 practitioners. Among the respondents, IT staff has the average rating in the range of 1.32 – 4.84. This is the lowest range compared to the other practitioners. The general ranking classification seems to suggest that all 4 groups of practitioners have common agreements for all of the 30 risk strategies. But a thorough examination revealed that IT staff has a contrasting ranking for a few strategies compared to the other practitioners. For example, strategies S4 (Proper project planning & scheduling), IT staff ranked the strategy as 3rd in its ranking list, with an average rating of 4.84. Other practitioners rated the S4 strategy with a slightly lower rank, which the Board of Directors ranked it as 12th, the Project Manager (8th) and Developer ranked S4 as 11th in their respectively ranking list. Despite the difference in ranking, these 3 practitioners (Board, PM & Developer) rated the strategy S4 with an average rating of 5.02, 5.04 and 5.11 respectively.

This shows that the IT staffs were more concern with the proper planning and scheduling of project, compared to the other practitioners. The nature of their work as being towards the end of the hierarchy and responsibilities line behind the Board, the PM and the Developers, would probably putting more pressure and work strain on them, if the project did not go as planned. As for strategy S8 (identify success criteria), IT staff also has a different opinions compared to the other 3 practitioners. IT staff rated this strategy as 2nd ranked in its ranking list, while Board (7th), PM (7th) and Developer (10th) in their respectively ranking list. Again, the IT staff tends to rate the success criteria factor higher than any other practitioners, seemed to suggest that IT staff would wanted to know exactly what is expected end product and how does the success of the project going to be measured.

IT staff also rated this 4 strategies, timeframe for testing (S17), planned for parallel or phased conversion (S19), backup the system (S22) and cost control procedure (S24), differently compared to the other 3 practitioners. This might be due to the expectation that IT staff is normally has less responsibility and liabilities from the perspectives of project management, but higher responsibility in terms of the task in hand that they need to do. The IT staff is usually expected to do most of the technical and direct hands-on task, and reported back to the Project manager or Developer. This may include activities such as testing, conversion of the old software to the new one and backing up the system. So, the IT staffs see these 4 strategies quite differently compared to the other 3 practitioners. Despite these differences the nonparametric test using Kendal's coefficient confirmed that there was a general agreement among the respondent at least for the ten top risk mitigation strategies.

Risk mitigation strategies are procedural actions aimed at reducing threats to the success of project by reducing their likelihood of impact of occurrence and impact (Bannerman 2009). Hence, identifying common strategies that might help in minimise project cost and time overruns is crucial the project management team. In Table 8.3, the research have found that there were common agreements among the practitioners in evaluating the effectiveness of risk strategies. The strategies that deemed of value in reducing the risk are users' involvement (S30), good project management (S29), lines of communications (S28), planning of resources (S26) and developed a clear systems requirements (S20) risk mitigation strategies. These 5 strategies were rated with a minimum average rating of 4.80 and a maximum average rating of 5.31, which means highly effective strategies.

This finding suggests that it is not so much about the technological importance of the project but also about the relationship of the people involved within the project, and those affected by the project. This finding confirms the view expressed by Shih-Chieh Su et al (2008) on the pivotal role of users' involvement in the development of software project. However, Barki et al (1993) and Jiang et al (2002) argued that it is difficult to understand and to predict the users' expectations and requirements, and thus the completion of the final project within the timeframe and the budget allocated. Although most literatures stressed the importance of systems requirements for the outset of the project, but the final or absolute version of the systems requirements would probably become clearer as the project progresses. The project managers should consider ensuring that project requirements are accepted by the team as being realistic and achievable, given the available time, resources and technology. Project managers must be willing to draw a line between desirable and absolute necessary functionality.

Project management literature also suggested the importance of the communications between the development team and the users in defining the project scope and controlling the project changes. The project managers and the development team need to build, create and maintain good relationship and trust with the users. Good project management in terms of how well work is planned, how well progress is controlled against plan, and how effective change control is, and whether a formal risk management processes are used is essential in the success of software project development. These results don't correspond with previous findings that suggest there is little evidence to indicate that project management is necessary for project success (Shenhar and Dvir, 2007). The research findings, however, correspond with the findings of (Bannerman 2009) that good project management is necessary but not essential in its own in mitigating project risks. Another explanation of the research finding could be attributed to the fact that project management is associated with effectiveness of project governance. Thus, the respondents' high rating of this risks mitigating strategy. This view collaborates with the observation made by (Bannerman 2009) "Absence of effective governance resulted in risk exposures in these areas (i.e., clarity and relevance of objectives, scope and requirements; provide guidance, direction and a common sense of purpose)"

As suggested in several studies communication issues among human being were relatively complex and unpredictable. Jiang & Klein (2000) stressed that, as a result of poor communications among development team members, much of the time might be spent on duplication of efforts and progress will be towards individual's goal rather than the project goal. Proccaccino et al (2005) also highlighted the importance of actively nurturing effective communication that improves interpersonal relationships of their team members. As well as project management expertise, project leaders should also have communication and people skills in managing the software development project. The importance of communication and participation of project stakeholders in reducing project risks reinforced by the findings from Bannerman (2009) that integration of teams has lead to "greatly improved project communication, interaction, issue resolution and progress tracking". These views collaborate with the findings in Table 8.3. For example, risk management strategies of S1, S4, S5, S7, S8 were rated quite high in the ranking. In fact all of these strategies were rated in the top 10 by all 4 groups of respondents, with an average rating of 4.76 to 5.11 which means highly effective strategies. These risk management strategies include strategies such as defining clear objectives, identifying critical activities, specifying the project success criteria, consistent commitment from management and also the lessons learned from past software projects. Tesch et al (2007) also mentioned the important of lessons learned from past project or projects recently completed as risk management strategies, to better perform future projects.

For a successful project, normally a considerable time is spent on the planning phase and sometimes longer than it takes to complete the rest of the project itself (Taylor,2003). The more common practice is to develop a plan, put it on the shelf and never look at it again until the project is finished. It is important to remember that the success of a project starts with the project plan. The initial project plan needs to be as thorough and detailed as possible. Determining what to do, how to do it, when to do and who should do it, is probably simple in concept but not always easy to accomplish. Successful project completion requires detailed and meticulous planning, careful monitoring of the state of project against the plan and the allocation of resources. Management need to consider the difficulties of estimating the manpower resource required, planning and coordinating the work of staff, and ensuring that the overall project is completed on time. System development projects are notorious for over running on time and cost budgets. A slippage at any particular phase cannot always be corrected by simply putting more staff or allocating more money at the problem. Possible responses to problems and slippages include maintaining the scope of the projects and carrying on with the same schedule, changing the scope of the project or even probably changing of schedules and resource allocation.

Although, previous studies suggested that there were quite a number of risk management tools or techniques available in reducing the risk, but in this study, risk management technique (S11) or tools was not considered as highly effective strategy. This strategy was rated with an average rating of 3.92 – 4.68. In fact, all 4 groups of respondents just barely rank this strategy in the top 15 out of the 30 risk management strategies. From Table 8.3, it can also be seen that strategies S10, S12 and S19, were consistently rated in the bottom of the rank. This includes strategies such as quality control procedure, hiring of external consultant or expertise and undertaking a parallel or phased conversion. These strategies have an average rating of below 2.0, which mean not an effective strategy. Hiring of external expertise may not be on most of the respondents' agenda, probably because most companies have enough human resources and expertise to get them through any particular software project. However, this finding don't correspond with the findings from other studies that suggest some project software development companies use risk transfer techniques to outsource the technical risk associated with a project (Bannerman 2009). Despite the quality control procedure being ranked lower in ranking, the respondents or companies would probably did not realize that they might already have the quality control procedure in place, but it is not formally known as such. It might be formed part of the project management functions such as coordination, monitoring or control. Speculatively it can be argued that quality control procedure that may have many formats such as formal or informal method and can include methods such as Capability Maturity Models (CMM), the Boehm's spiral model or some form of quality management metrics.

8.9. Summary

This chapter reports the findings from an empirical survey on the effectiveness of the risk management strategies in a software development project. From the study, it is clear that the respondents viewed the non technical related strategies were more effective than the technical ones. The analysis shows that risk management strategies relating to users' involvement, project management and planning and communication issues are considered very influential on reducing the effect of the risk towards the software projects. Even the lessons learned from past software project was considered as highly effective risk management strategy.

In software risk management strategies, course of action could include a straight forward solution on the risk, or it could involve avoiding the risk by eliminating its cause. Much of the best approach to software risk management strategies is to anticipate those risks and taking some action in advance to ensure that either the potential effect of the risk is reduced, or its likelihood of occurrence is reduced, or both.

Although it is useful to see these strategies from the point of view of understanding the approaches to a risk situation, it might not be wise to try to pigeonhole any practical approach into one or other category in an exclusive way.

CHAPTER 9 FUZZY TECHNIQUES OF RISK ANALYSIS IN SOFTWARE DEVELOPMENT PROJECT

9.1. Introduction

The effectiveness of the evaluation and estimation processes can provide valuable support for decision making of any project. The processes can involve quantitative or qualitative analysis depending on the information available and the level of analysis that is required. Quantitative techniques rely heavily on statistical approaches, and qualitative techniques rely more on subjective judgement. Both quantitative and qualitative techniques have their own advantages and disadvantages. However, usually incomplete project information is available during the very early phases of the project and many decision making processes occur in an environment in which the goals, constraints and consequences of possible actions are not precisely known. Qualitative issues such as behavioural, political and other organizational concern are becoming more critical to project success than ever before.

In order to improve the decision making process with the lack of information available, several mathematical programming models had been introduced and proposed, like multiattribute decision making, dynamic and non-linear programming (Alexander et al, 1985; Schniederjans et al, 1989; 1993; Santham & Kyparisis, 1995; Badri et al, 2001; Chen & Tzeng, 2001). However, decision makers usually refrain from using such techniques or models due to complex programming and implementing processes. Furthermore, mathematical programming methods also need crisp and precise data to get meaningful results.

In evaluating the rating of risk factors, most decision-makers or project managers, are more comfortable viewing those factors as linguistic values, e.g., high, moderate, low, very low, likely, unlikely, minimum, maximum, etc.etc, rather than in exact probabilistic terminologies (Engel & Last, 2007; Engel & Shacher, 2006; Engel & Barad, 2003). However, sometimes, the scoring methods or ranking methods might have a compensatory bias. For example, when one criterion has a low value other criteria may offset it, and then a project with a high weighted score might be accepted. Any extreme low or high values could bring the average up or down, which could lead to misleading conclusions and interpretations.

Hence, subjective human ratings and evaluation process can be better approximated using fuzzy measures than using the additive ones. Fuzzy logic has been employed in handling inexact and vague information because of its ability to utilize natural language in terms of linguistic variables (Ghotb & Warren, 1995; Zeng & Smith, 2007; Chen & Cheng, 2009). The arithmetic and calculus of fuzzy sets and fuzzy numbers provide us with a method for manipulating these imprecise representations (Dubois & Prade, 1980; Chen & Gorla, 1998; Zadeh (1994). Through fuzzy, the decision making processes can still be modelled and justification for the decision can still be made eventhough with limited project information.

9.2. Software project risk model

As being highlighted in the early chapters, the general conclusion from the literature review is that the IS literature is a jumble of diverse risk models and partially overlapping lists of risk factors and components. This resulted in the lack of practical model or risk construct that most managers can use for understanding risks in software project at whatever detail is appropriate for them. The risk construct proposed in this research which is based on project management life cycle addressed this issue.

There are many risk analysis techniques being used and introduced, such as quantitative methods like Artificial Neural Network (ANN), Bayesian Belief Network (BBN) (Bennet et al, 1996; Ngain & Wat, 2005) and qualitative techniques such as risk management standards by professional institutions (IEEE, PMI, ISO). However, existing methods for risk assessment are largely based on more generic approach of risk checklists and analysis of a risk matrix (Xu et al 2003; Down et al, 1994; Zhiwei Xu, 2001; Costa et al, 2007; Iversen et al, 2004; Jiang et al, 2001). Various risk factors are usually scored in the risk matrix according to their influence on the risk, and then arithmetically aggregated into an overall risk score.

The risk checklists are usually compiled from surveys of the experienced of stakeholders who had been involved in software project. The risk management value of this technique is that the factors may also be important or generic risks for other projects. Furthermore, most researches also proposed a variety of categories and frameworks according to related themes and characteristics. These frameworks may provide broader framing for thinking about what risks might threaten a particular project, rather than to simply work through a pre-defined checklist of specific factors.

However, this risk checklist and framework techniques need to be used with extra care as they are unlikely to be universally applicable. The best use of this technique is as a starter list in evolving a customised in-house set of risk factors from the software projects conducted in the organization over time. Factors that are not relevant can be omitted and new factors can be added. This is where this research risk construct based on project management life cycle contributed in solving the problem of risk checklist. Although initially, the research used the risk checklist as the framework, but the risk lists were based upon extensive literature reviews of risk in software project. In addition, the stages within the project management life cycle would allow the risk lists be updated relevance to the related process or activities of the life cycle.

Since, the project management life cycle perspectives was used for the research risk construct, the clearly defined stages of the life cycle should be understandable to typical business managers and not necessarily only risk managers or IT managers. With this construct, it should provide rigor by organizing risk management analysis, without sacrificing practicality where businesses and organizations can include or exclude specific risks to the organizations situations and interests. Another added contribution from this research construct is that the risk lists used for this study were surveyed to the main software practitioners within the software project. Unlike other published lists which usually based on the perceptions of a single stakeholder group, predominantly project managers and users. It is critical that the views of all main stakeholder groups are taken into account, as (March & Shapira, 1987; Schmidt et al, 2001; Bannerman, 2008) pointed out that the stakeholders groups tend to identify risks in the responsibility domains of other stakeholders, rather than point to factors as risks within their own area of responsibility.

Some other software project risk model being proposed and developed include contingency model (Barki et al, 2001), socio-technological model (Lyytinen et al, 1996), options model (Benaroch, 2002), Bayesian Belief Network (BBN) model (Kwok Tai Hui & Biau Liu, 2004) and Neural network technique (Neumann, 2002).

The contingency model (Barki et al, 2001) focused on the software development project performance is influenced by fit between risk exposure and risk management. But the model did not organized the risk factors into specific framework and did not distinguish between initial and emergent risks. Distinguishing between risks that exist prior to a phase and those that emerge during a phase is important for risk management because those factors may be managed differently. However, categorizing the risk factors based on the life cycle stages framework as proposed for this research would allow the initial risks factors being identified earlier in the life cycle, and the potential emergent risk factors being identified in the later stages. The socio-technological model (Lyytinen et al, 1996) classifies risks by system, project and management sources. But, the sociotechnical model of organizational change only highlighted on components internal to the organization such as task, structure, stakeholders and technology, and mainly during a software project's development phase only. Unlike the framework used for this research where it covers the whole project life cycle.

Benaroch (2002) presents an approach that uses real options approach (deferral, piloting, outsourcing, abandonment, and so on) to actively configure IT investments for the purpose of managing the balance between their value and risk. More precisely, building on the notion that real options can control IT investment risk. It considers risks arising outside the scope of development. But, its mapping of risk to specific options with a high level of generality makes it difficult to use for identifying risks. It addresses investment risk across a sequence of choices. It includes positive and negative financial outcomes but does not identify other sources of risks in addition to specific competitive, market and organization specific risks. Kwok Tai Hui & Biau Liu (2004) use the Bayesian Belief Network (BBN) model to develop a scientific tool that can be used to understand and calculate the risks of a software development project. The model assesses values that are critical to a project by calculating their associated risks and probability of their occurrence each with a weight factor to derive their impact. However, the model only focused on factors that may go wrong at the beginning of a typical software development project and not for the whole project life cycle.

The artificial neural network technique (ANN) being used in (Neumann, 2002) to categorized the risk factors with the utilization of principal component analysis (PCA). The approach draws on the combined strengths of pattern recognition, multivariate statistics and neural networks. Principal component analysis is utilized to provide a means of normalizing and orthogonalizing the input data, thus eliminating the ill effects of multicollinearity. A neural network is used for risk determination/classification. A significant feature of this approach is a procedure, termed cross-normalization. This procedure provides the technique with capability to discriminate data sets that include disproportionately large numbers of high-risk software modules. However, a more complete assessment of software metrics and their potential for use in risk classification is needed for the model to be applied across the organization. Several neural network approaches were also need to be used to classify the data sets. The data used in the classification analysis was primarily from one organization. Additional data from other organizations would be beneficial for further justification of the model application.

Other common technique or models normally used for risk management are the software risk management models or standards proposed by professional institutions like PMI risk management model, IEEE risk management process and ISO risk management guidelines. However, as their names might suggest, these techniques are more like guidelines and standards, rather than specific modelling techniques. These are often interrelated and used together with the checklists and framework approach explained previously. Typically, most of these standards specify the individual activities necessary to manage the risk in software projects. For example, stepwise tasks like risk identification, risks analysis, risk response and risk control. The ordered steps are usually intended to be executed iteratively throughout the project, to manage known and new risk factors as the projects proceeds (Simister, 2004, Bannerman, 2008). The major contribution of these standards are that they guide and direct risk management actions. However, these models require skills, judgement and persistence to effectively apply them in practice. For example, having identified and analyzed the risks, it is then necessary to determine what and how should be done with the factors.

9.3. Why use fuzzy modelling

Traditional risk assessment methods usually model risks as objective probability based on expected frequency of repeatable events with regular usage of linguistics words to represent the likelihood and impact (Zhiwei et al, 2003). However, the exact or discreet values of probability of occurrence and impact on cost overrun cannot be given to risk, at least not in real terms because of the uncertainty and vagueness of risk factors (Shull, 2006). Since risk has no exact value, traditional quantitative risk assessments are usually qualified with a statement of uncertainty and relatively intangible in nature.

Most of the risk assessment techniques tend to ignore vagueness, nonspecificity, fuzziness, and ambiguity that are due to the lack of specific information, ignorance, scarcity of data, and conflicting information (Altunok et al., 2006; Jin & Doloi, 2009). According to Thomas et al. (2006), because most of them are basically probability-oriented, they do not identify all the factors necessary to reflect realistic situations and cannot cope with a problem bearing complex relationships among various variables. In addition, very little is known about software project in the very early phases of software development, and thus, probabilistic models usually become impractical. Therefore, in such early phases it is a challenging task to show correlations between the limited information available and the problems that may arise in later phases of the software development life cycle (Zhiwei et al, 2003). Furthermore, in the early stages of the software development life cycle, it is probably difficult to guarantee the choice of assessment scales and that their weighting factors are fair and realistic.

In this study, the research illustrated how the fuzzy theory model suitable for solving imprecise and subjective problems, in contrast to the traditional risk matrix based assessment technique. The proposed model offers a quantitative value of the risk factors, because the decision of accepting the risk is taken by the human (project managers, IT managers or other stakeholders) (Bodea & Dascalu, 2009). The output of the model is not a form of decision, but an important parameter to make a proper decision. In a probabilistic approach, the imprecise and uncertainty were modelled by expressing the belief that an event either occurs or does not, but in contrast, fuzzy-logic membership functions express the possibility of an outcome rather than the likelihood of an outcome (Darbra et al, 2008). With fuzzy logic, the uncertainty was also modelled as the degree of membership in the set that defines an outcome (McKone & Deshpande, 2005)

The key advantage of fuzzy-logic membership functions compared with traditional mathematical models lies in the fact that relationship between inputs and outputs is not determined by complex equations, but by a set of logical rules, reflecting an expert's knowledge (Gonzalez et al, 2002). The algorithm created is based upon fuzzy logic, giving this the ability to solve complex problems plagued with uncertainty and vagueness. This uncertainty can makes stakeholders nervous about investing in a new project, which makes it imperative to analyse these risks, but not in the traditional way where specific values are given to the probability of risks to occur and their impact; but in a new way where the stakeholder has a margin of error that will not affect the analysis (Shull, 2006).

Fuzzy logic was used since it is a tool capable of modelling complex and uncertain or vague using simple terminology (Bodea & Dascalu, 2009). Fuzzy logic provides a quick and efficient tool for project managers in their use of project evaluation, by allowing the project managers to assess the risks factors without putting the least amount of effort into an analysis.

Managing risks should involve making decisions based on the information collected in risk assessment. However, it is important to remember that the final decision of how to manage is generally human in nature.

The flexibility of fuzzy logic to express results in a natural language, in line with human reasoning, together with the possibility of dealing with uncertainties makes it highly recommended as a tool for use in communicating risk (Darbra et al, 2008; Adeli and Sarma, 2006; Han and Diekmann, 2004).

9.4. Fuzzy in software project risk

Since Fuzzy set theory (FST) was introduced by Zadeh (1965) to deal with problems in which vagueness was present, fuzzy theory is finding wide popularity in various applications that include management, economics, and engineering (Boussabaine & Elhag, 1999; Vakili-Ardebili & Boussabine, 2007)(Ying-Ming & Elhag, 2007, 2008). The linguistics values and variables have been widely used to approximate reasoning.

Fuzzy techniques are not limited to a specific field or area of science. It can be used by any branch of science which involves uncertainty and subjectivity. Fuzzy logic application embraces numerous areas. Several researches applied and developed fuzzy rules for different purposes. Some research have also used fuzzy logic and fuzzy rule based modeling in their risk assessment methods in different fields such as construction projects (Dikmen et al, 2007; Ying-Ming & Elhag, 2007, 2008), petroleum projects (Roisenberg et al, 2008), human health (Shakhawat et al, 2006) and safety (Nait-Said et al, 2008).

For example, Engel and Barad (2003) proposed a set of quantitative probabilistic models for estimating costs and risks of software Verification, Validation, and Testing (VVT). Engel and Last (2007) extended that research by modelling the software testing risks problem using the fuzzy logic paradigm. Their research started of with the assumption that the software failures are mainly contributed by the ineffective performance of software and systems Verification, Validation and Testing (VVT). The research modelling approach was focused on calculating costs and risks stemming from carrying out a given VVT strategy, while making comparisons between probabilistic models and fuzzy modelling. They concluded that their models will provide management with a decision support tool to evaluate proposed testing alternatives. However, their research is only concerned with the modelling of the VVT at the phase of testing specifically, but not for the whole life cycle of IT projects.

Lee (1999) builts a hierarchical structure model of aggregative risk in software development and rated aggregative risk in a fuzzy environment by fuzzy set theory. They classified each risk item into two fuzzy sets with triangular membership functions: grades of risk, grades of importance, and rate of risk. In succeeding studies Lee et al (2003) then evaluated the rate of each individual risk item using a two-stage fuzzy assessment method within a group decision making settings. They have used 13 linguistic values. Eventhough, they stated that their modelling can be used during any phase of the life cycle, but, the method of the two stages fuzzy assessment makes the modelling fairly complex and time consuming. They also use a risk pairing of grade of risk and grade of importance of risk, rather than the risk factors individually. In contrast with this research, the simple and straight forward fuzzy modelling proposed for in this research should make the model easier and more understandable for interpretations.

Zhiwei et al (2003) has also developed a fuzzy expert system to support assessment of the operational risk of software during operations due to software failures. Their study used the fuzzy expert rules of "IF-THEN", generated from the experts of the software engineering fields. They used the fuzzy joint probability by applying the Bayes Theorem. Besides the number of rules that have to be generated, the study only focuses on risk factors that come to occurrence when the software system is implemented and fully operational. Their research focus on operational risk factors such as technology risk, risk of poor replan, software developers competence and also project risk. It did not consider the whole development and project management processes of the software development life cycles.

Ngai & Wat (2005) describes the research and development of a fuzzy decision support system (FDSS) to assist E-commerce project managers in identifying potential risk factors and evaluating the corresponding E-commerce development risks. This FDSS proposed would help in the evaluation of a company's risk level and provides an overall risk evaluation of E-commerce development. However, since this proposed FDSS is a web-based design, the focused were more on risk identification and risk analysis. Less attention was given to the risk management planning and monitoring. In addition, risk monitoring should be conducted regularly in tracking the status of the identified risks. Moreover, it was assumed that the weighting assigned by each evaluator in the risk evaluation was the same, but the relative importance placed on certain factors by individual decision makers and experts could be widely different. In spite of the prototype evaluation shows a satisfactory outcome, but the validity of the system need to be established through in-depth case studies of real-life E-commerce projects.

Iranmanesh et al (2009) developed a two-layer fuzzy expert system to evaluate categorized risk factors and the total risk of software projects as a decision support tool. A risk assessment fuzzy expert system was developed to evaluate the risk of software projects where the risk factors were categorized into twelve categories. The expert system used a rule base with about 17 million rules. Instead of constructing the whole rule base, a heuristic programming was created to infer the inputs without losing any rules. The output of the model is numerical values which present state of risk for each factor as well as the risk of project called the total risk. The proposed tool may be used as a decision support system for top management to compare different projects or better risk mitigation in these projects. But the high number of rule base required and the complexity of the two-layer fuzzy membership could limit the application of the system.

While all these models are all useful to business managers and organizations, each of them is limited to a particular aspect of risk analysis. The abovementioned studies of fuzzy logic of risk in software project only consider part of the software life cycle process. The focused is either in the beginning of the life cycle, or for one specific phase like testing, or only for project selection processes. Most of the research also used rule based reasoning like the "IF-THEN" rules and complex programming and algorithm. They also do not include the dynamics of risk in the broader aspects of projects or systems in general. These models are inadequate for describing, analysing or communicating the range of risks that are relevant to software projects because many of these risks are business and organizational risks that are often considered beyond the scope of software development (Susan & Alter, 2004). The framework or model based upon project management perspectives life cycle proposed with this research may be more useful for analysing the risks and for communicating with other stakeholders within the organization. At the same time, this will allow the team to assess the risk of software development project in more systematic manner.

9.5. Methodology

Questionnaire designed and developed in this study is used to collect different individuals' points of view on risk factors. The software practitioners based on their personal experience and knowledge ranked the risk factors. The results of the survey and analysis in the factor analysis were used for the fuzzy modelling. Through the factor analysis process, 45 risk factors were extracted to be most influential risk factors for the likelihood occurrence and risk factors impact on cost overrun of the software development project.

Although fuzzy theory deals with imprecise information, it is based on sound quantitative mathematical theory (Chen and Hang, 1992). A variable in fuzzy logic has set of values which are characterised by linguistic expression such as high, medium, low, etc, etc. Linguistic variables as described by Zadeh (1994) provide a means of modelling human tolerance for imprecision by encoding decision relevant information into labels of fuzzy sets. A variable in fuzzy logic has set of values which are characterised by linguistic expression such as high, medium, low, etc, etc. These linguistic expressions are represented numerically by fuzzy sets.

The method which is employed in this research is based on application of fuzzy techniques with linguistic variables to represent risk factors indicators. Dubios and Prade (1980) developed an approach taking into account the weight of each risk factor using the formulas :-

$$\mathbf{F}(\mathbf{y}) = \sum \mathbf{W}_i \mathbf{F}_i(\mathbf{x}) \qquad \text{for sum } W_i = 1 \qquad \text{- formula 1}$$

Extraction of the Membership Function (MBF) of the sets is the most important aspect in a fuzzy decision support systems development. There are many guidelines on developing the membership functions for fuzzy sets (Dubios and Prade, 1980).

Each fuzzy set carries a distinct membership function belonging to the interval 0 to 1, Degree of membership varies from 0 (non-membership) to 1 (full membership). This is in contrast to crisp or conventional sets, where an element can either be or not be part of the set (Boussabaine and Elhag, 1998). Furthermore, Dong and Wong (1985) developed an approximation technique that applies alpha cuts (alpha) (a horizontal line which creates cross-section at the level of membership). MBF for each risk factors is calculated based on the estimated alpha cuts (alpha), and then an average weighted membership based on MBF of its factors is computed using the following equation to compute the final score of each category of risk factors.

$$\mathbf{Fi\ (y) = \sum W_i\, F_{ij}\ (x)\ /\ \sum W_i} \qquad \text{- formula 2}$$

Where $F_{ij}(x)$ is the membership function at a certain alpha cut (alpha)

And W_i is the weighting coefficient for criterion (a)

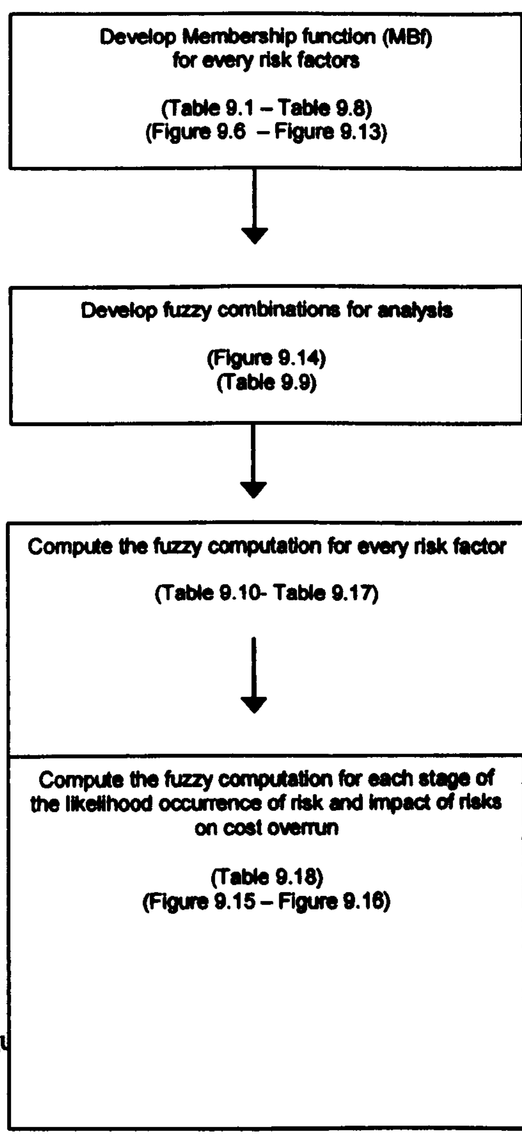


Fig [illegible] elling

In this research, the MBF for the risk factors is constructed based upon their statistical characteristics such as the average weighted mean and standard deviation. This is justified by Boussabaine and Elhag (1998) that the advantage of statistically based membership function is that they are naturally quantitative, which, there is reason to believe that the MBF has a relationship to some physical characteristics of the set.

After constructing the Membership function (MBF), the linguistic variables is defined. The real value of the risk factors is transformed into a linguistic value through the application of linguistic variables. As mentioned previously, linguistic variables could be in the form of non-numeric quantities, terms such as Low, Moderate, High or Minimum, Maximum, etc. The variables can be described in more detailed terms, as there's no limitation on the number of terms that can be used. Since the calculations of the linguistic variables and values quite time consuming, therefore, only three (3) linguistic terms is used for this research, that is, Low, Moderate and High.

9.6. Development of fuzzy Membership function (MBF)

In order to define a representative membership function, there are conditions which can be imposed to make the set have characteristics consistent with the subjective judgement of the decision maker. (Civanlar and Trussell, 1986; Boussabaine and Elhag, 1999). For this research, the mean and standard deviation of the risk factors scores were used to develop memberships of functions of risk factors. The fuzzy membership function (MBF) of x is defined as F (x) which belongs to [0,1], are estimated by using the following formulae (Boussabaine and Elhag, 1999; Vakil-Ardebilli & Boussabaine, 2007) :-

i. For low level of significance of a defined risk factor

$$F(x) = |(a-x)/b| \quad \text{for} \quad a-b < x < a$$

ii. For medium level of significance of a defined risk factor.

$$F(x) = |(x-a+b)/b| \quad \text{for} \quad x < a$$

$$F(x) = |(x-a-b)/b| \quad \text{for} \quad x > a \quad \text{or} \quad x = a$$

iii. For high level of significance of a defined risk factor.

$$F(x) = |(x-a)/b| \quad \text{for} \quad a < x < a+b$$

For this research, '***a***' is the average mean and '***b***' is the standard deviation. From the formula, it can be seen that, there is a focal central member, '***a***' for which F(x) is greater than other members of the set. Whereas, '***b***' is a controlling scale factor parameter. These parameter influences the shape and distribution of the equations, as shown in Figure 9.2. In Figure 9.2, the horizontal scale values represent the level of significance of a risk factor using mean and standard deviation. The fuzziness and MBF is increased or decreased by the parameters '*a*' and '*b*'.

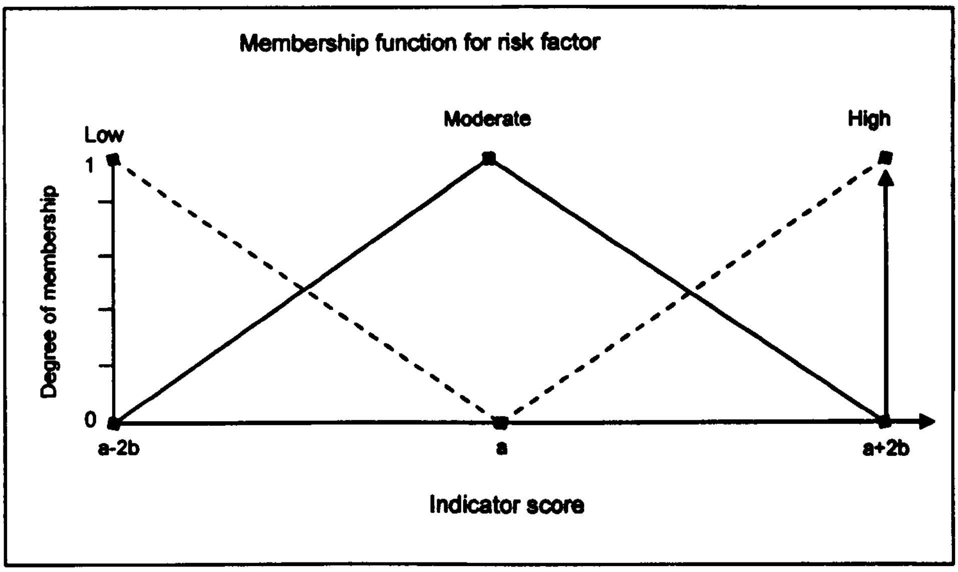


Figure 9.2 : Membership function of risk factors

Since the mean is an unbiased estimate for any sample set, it is an ideal choice for '*a*'. Figure 9.2 shows the range of moderate level of significance of risk factors from (a-2b) to (a+2b); with the highest degree of membership occurring at the value of (a-2b) for risk factors level of significance; '*a*' for medium level of significance; and at (a+2b) for high level of significance. Assuming the distribution is normal, it is usual for the distribution of the response data to lie within two (2) standard deviation away from the mean, and, with that in mind, the interval of (a-2b) and (a+2b) was chosen based on the survey data in this research.

9.7. Fuzzy Computation

The formula, presented previously is used for the assessment and calculation of the scores of the risk factors for the stages. In developing the scoring system, the value of the risk factors is transformed into a linguistic value through the application of linguistic variables. The high number of more linguistic variables can provide more scenarios and possibly better accuracy, but with higher complexity. These linguistic variables can be changed to suit particular project using relevant experience and knowledge from the project. However, for this research, the fuzzy approach is modelled by three (3) linguistic terms, that is, Low, Moderate and High category of responses. The linguistic weights are expressed in terms of degree of significance that the likelihood occurrence of risk factors will be Low, Moderate and High. This can be viewed as the probability of risk occurrence.

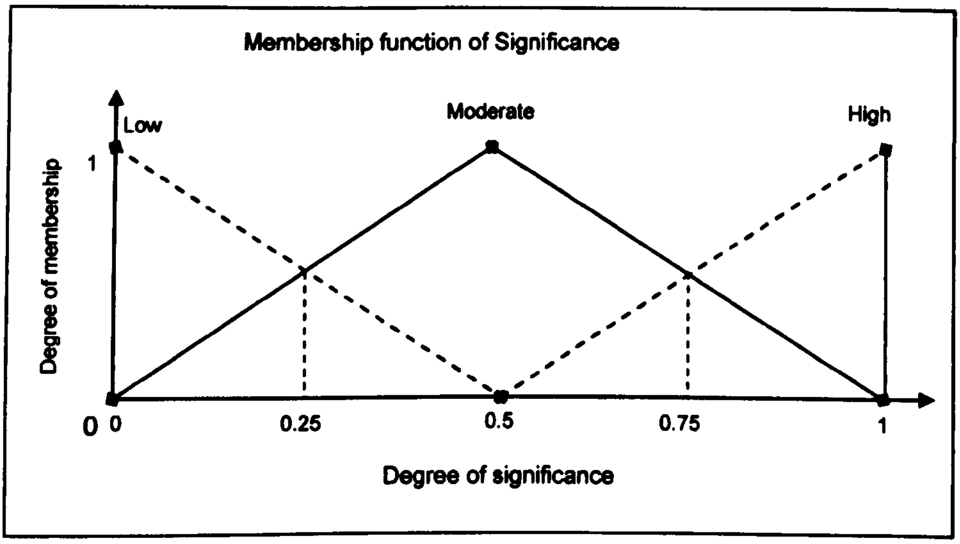


Figure 9.3 : Membership function of degree of significance for the risk factors

Figure 9.3 shows the membership function for the degree of weights (belief) that a particular risk factor will occur. Through Figure 9.3, the weights of each risk factor can be estimated and used in the Formula 1 and Formula 2 to compute the final score.

The process of fuzzy computation and combinations is demonstrated using the two triangles of MBf for risk factors and MBf of degree of significance. One triangle represents the MBF of a risk factor and the other one presents the MBF of degree of significance of risk factors (probability of occurrence) as shown Figure 9.4.

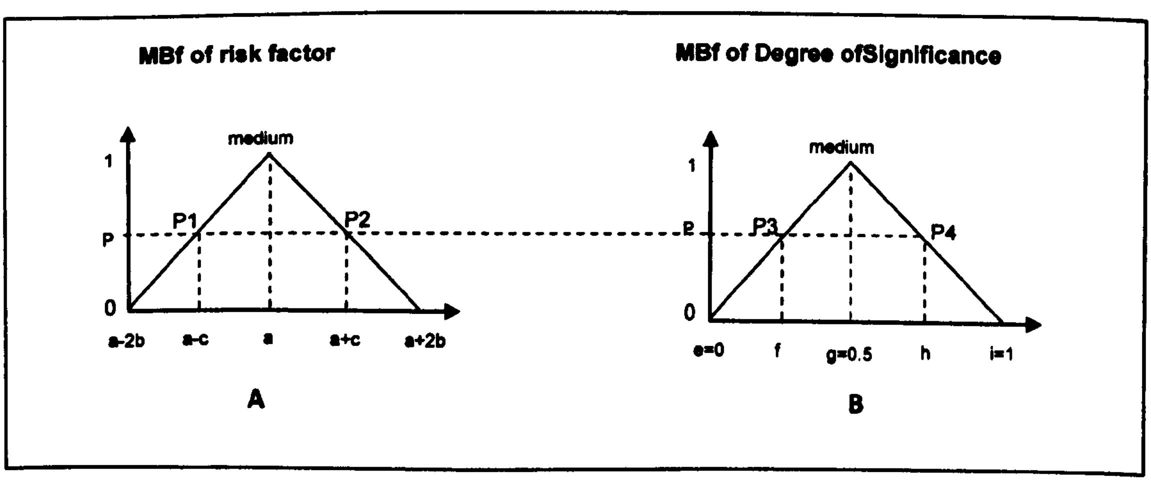


Figure 9.4 : Fuzzy combination process (Adapted from Boussabaine & Elhag, 1999; Vakili-Ardebili & Boussabaine, 2007)

Figure 9.4 above is used to illustrate the fuzzy combination process of the score and the weights by the concept of alpha-cut point (horizontal cross-sections at various levels of membership).

For example, from point P on the vertical axis (degree of membership, a horizontal line is drawn and point P shows the alpha cut point. This will create two intersections with the other two sides of triangle. These intersections in triangle A are shown as P1 and P2, whereas in triangle B are shown as P3 and P4. The extrapolation of P1 and P2 will have the scores of *(a-c)* and *(a+c)*. As for P3 and P4 in triangle B, the extrapolation will show the extracted weight of *(f)* and *(h)* relating to the score in triangle A.

The process of combination can be summarised as follows:-

Alpha cut at P = 0;	(a - 2b) * e;	and	(a + 2b) * i;
Alpha cut at P = P;	(a – c) * f;	and	(a + c) * h;
Alpha cut at P = 1;	a * g		

Therefore;

Sum $W_jF_{ij}(x) = \{(a - 2b) * e\} + \{(a + 2b) * i\} + \{ (a - c) * f\} + \{(a + c) * h\} + \{ a * g\}$

The example of fuzzy computation is shown in Figure 9.5.

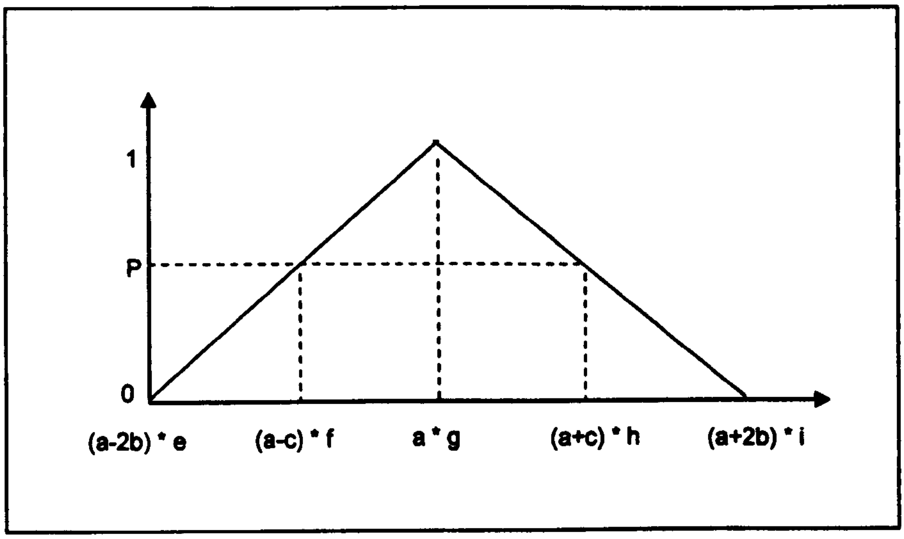


Figure 9.5 : Example of fuzzy computation

9.8. Data used in developing the Membership function (MBF)

As explained in the Factor analysis chapter, 45 risk factors were extracted for the Likelihood occurrence of risk, and 45 risk factors were extracted for the risk factors impact on cost overrun. Hence, these 45 factors were considered for the fuzzy membership function.

9.8.1. Membership function for Likelihood occurrence

The mean and standard deviation of the risk factors for the Likelihood occurrence is used for developing the membership function (MBF). The parameters 'a' and 'b', as explained in the fuzzy computation processes above, were used to obtain the membership function for various level of significance for each risk factors in every stage.

The data used for developing the membership function for the Likelihood occurrence of risk factors was shown in Table 9.1 – Table 9.6. The figures resulted from Table 9.1 – Table 9.6, were plotted in graphs form to illustrate the MBF of likelihood occurrence for risk factors of the stages. This illustration is shown in Figure 9.6 – Figure 9.11.

Table 9.1 : Feasibility stage data of the Likelihood occurrence

Stage	Risk factor	Average mean, (a)	Standard deviation (b)	a – 2b	a - b	a + b	a + 2b
Feasibility study	F1: Inproper justification of cost benefit analysis	3.6	0.94	1.72	2.66	4.54	5.48
	F2: Too narrow focus on the technical IT issues	3.249	0.886	1.477	2.363	4.135	5.02
	F3: Overlooked the management and business impact issues	3.47	1.06	1.35	2.41	4.53	5.59
	F4: Wrong justification of investment alternatives and opportunity cost	2.28	0.835	0.61	1.445	3.115	3.95
	F5: Inappropriate technology chosen from the feasibility study	2.44	0.862	0.716	1.578	3.302	4.164

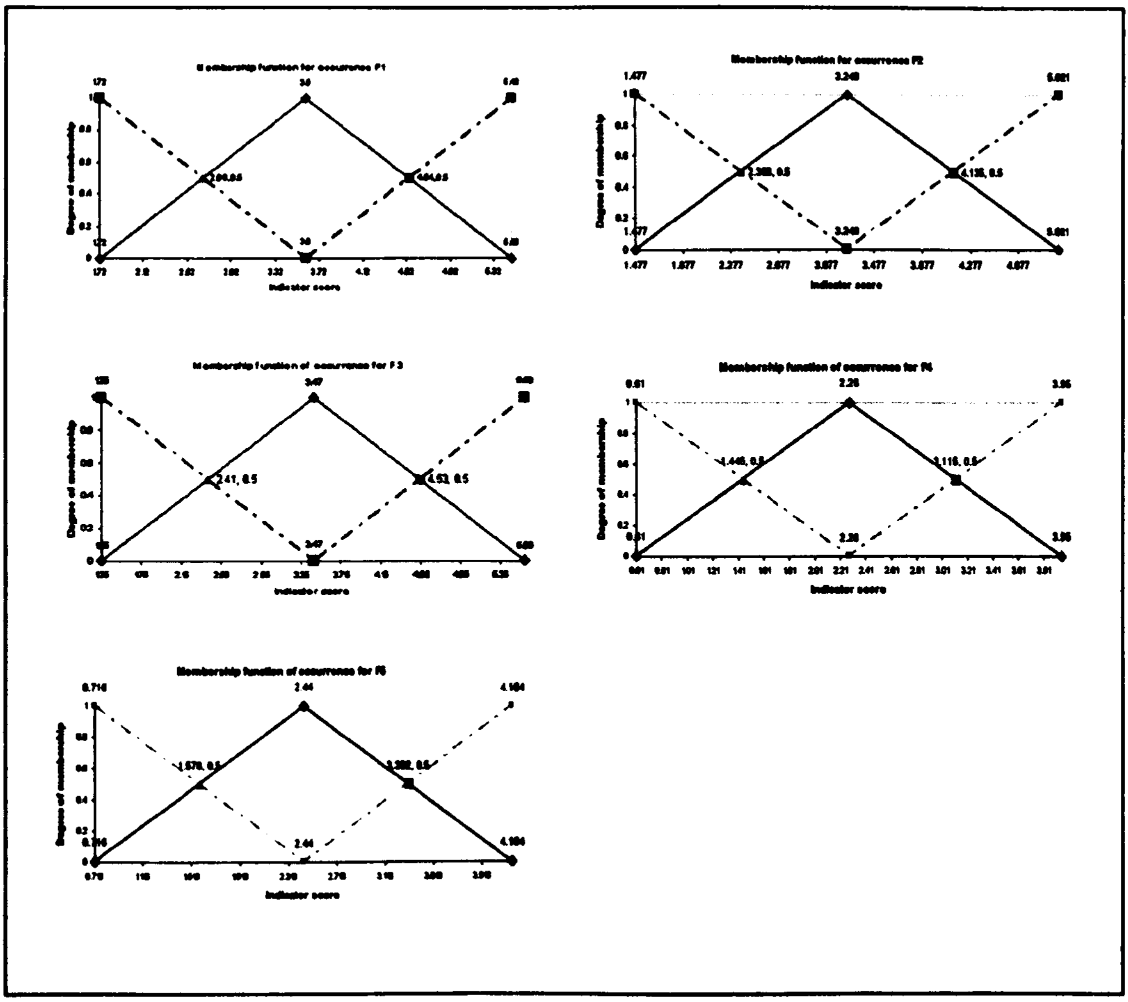


Figure 9.6 : Membership function for Feasibility stage of likelihood occurrence

Table 9.2 : Project planning data of the likelihood occurrence

Stage	Risk factor	Average mean, (a)	Standard deviation (b)	a – 2b	a - b	a + b	a + 2b
Project Planning stage	P1: Unclear project scope & objectives	4.17	1.07	2.03	3.1	5.24	6.31
	P2: Undefined project success criteria	3.69	0.829	2.032	2.861	4.519	5.348
	P3: Lack of quality control procedure and mechanism	2.72	0.964	0.792	1.756	3.684	4.648
	P4: Project milestones for stages not well established	2.41	0.829	0.752	1.581	3.239	4.068
	P5: Inproper change management plan	3.56	0.976	1.608	2.584	4.536	5.512
	P6: Inaccurate estimate of resources	4.44	0.854	2.732	3.586	5.294	6.148
	P7: Unrealistic project schedule	4.41	0.784	2.842	3.626	5.194	5.978
	P8: Inadequate detail work breakdown structure	2.56	0.986	0.588	1.574	3.546	4.532
	P9: Critical & non-critical activities of project not identified	3.73	1.019	1.692	2.711	4.749	5.768
	P10: Project management & development team not properly set up	1.9	0.94	0.02	0.96	2.84	3.78
	P11: Unclear line of decision making authority throughout project	3.99	0.897	2.196	3.093	4.887	5.784
	P12: Lack on contingency plan/back up	3.66	0.615	2.43	3.045	4.275	4.89
	P13: System conversion method not well planned	3.325	0.94	1.445	2.385	4.265	5.205
	P14: Improper planning of timeframe for project reviews and updating	3.187	0.821	1.545	2.366	4.008	4.829

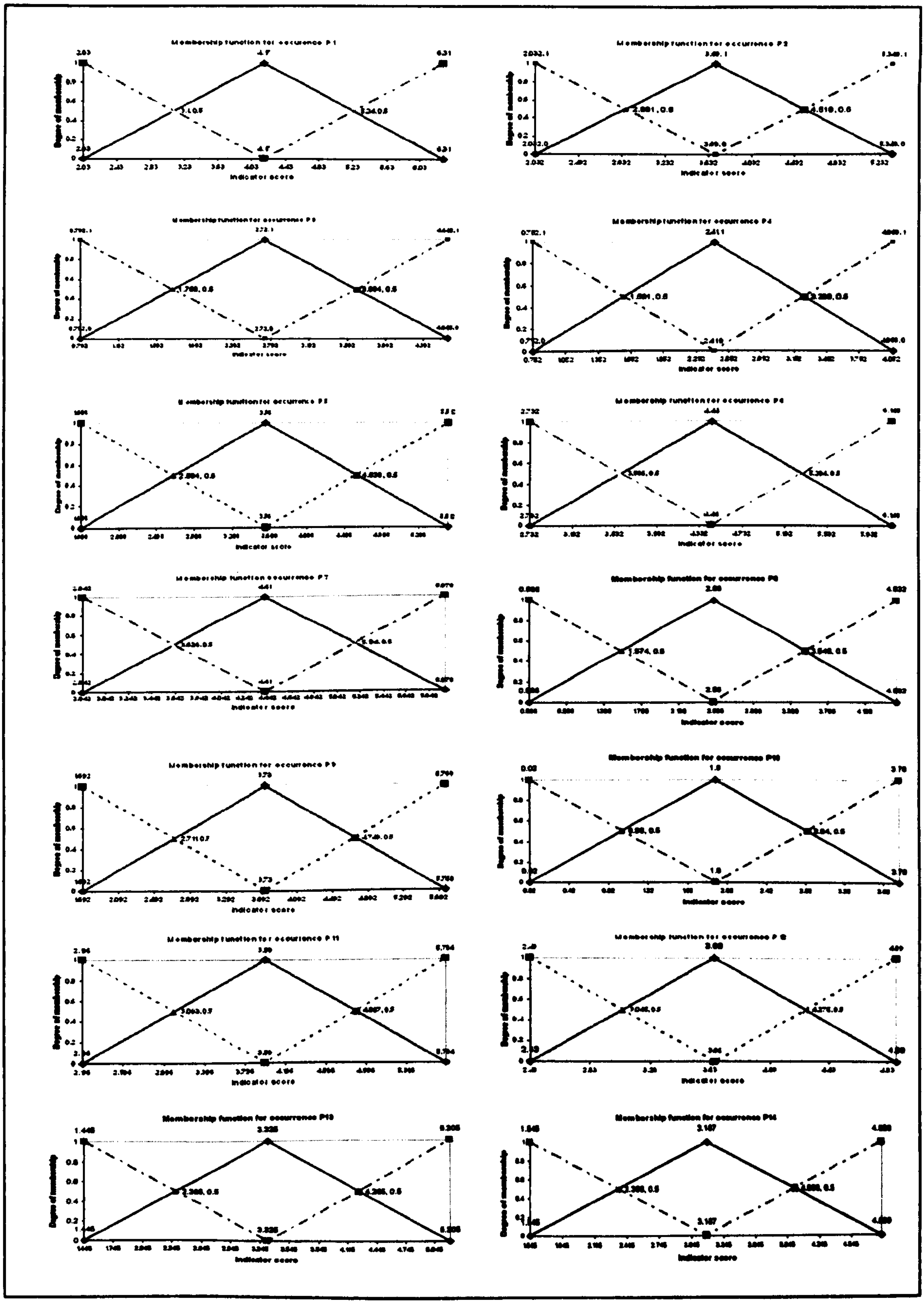


Figure 9.7 : Membership function for Project planning stage of likelihood occurrence

Table 9.3: Requirement stage data of the likelihood occurrence

Stage	Risk factor	Average mean, (a)	Standard deviation, (b)	a – 2b	a - b	a + b	a + 2b
Requirement stage	R1: Unclear & inadequate identification of systems requirements	4.08	0.767	2.546	3.313	4.847	5.614
	R2: Incorrect systems requirements	2.3	1.1	0.1	1.2	3.4	4.5
	R3: Misinterpretations of systems requirements	3.32	0.852	1.616	2.468	4.172	5.024
	R4: Conflicting system requirements	2.44	0.986	0.468	1.454	3.426	4.412
	R5: Gold plating or unnecessary functions and requirements	2.05	1.02	0.01	1.03	3.07	4.09
	R6: Inadequate validation of systems requirements	3.82	0.974	1.872	2.846	4.794	5.768
	R7: Lack of users involvement in requirement stage	3.41	1.658	0.094	1.752	5.068	6.726

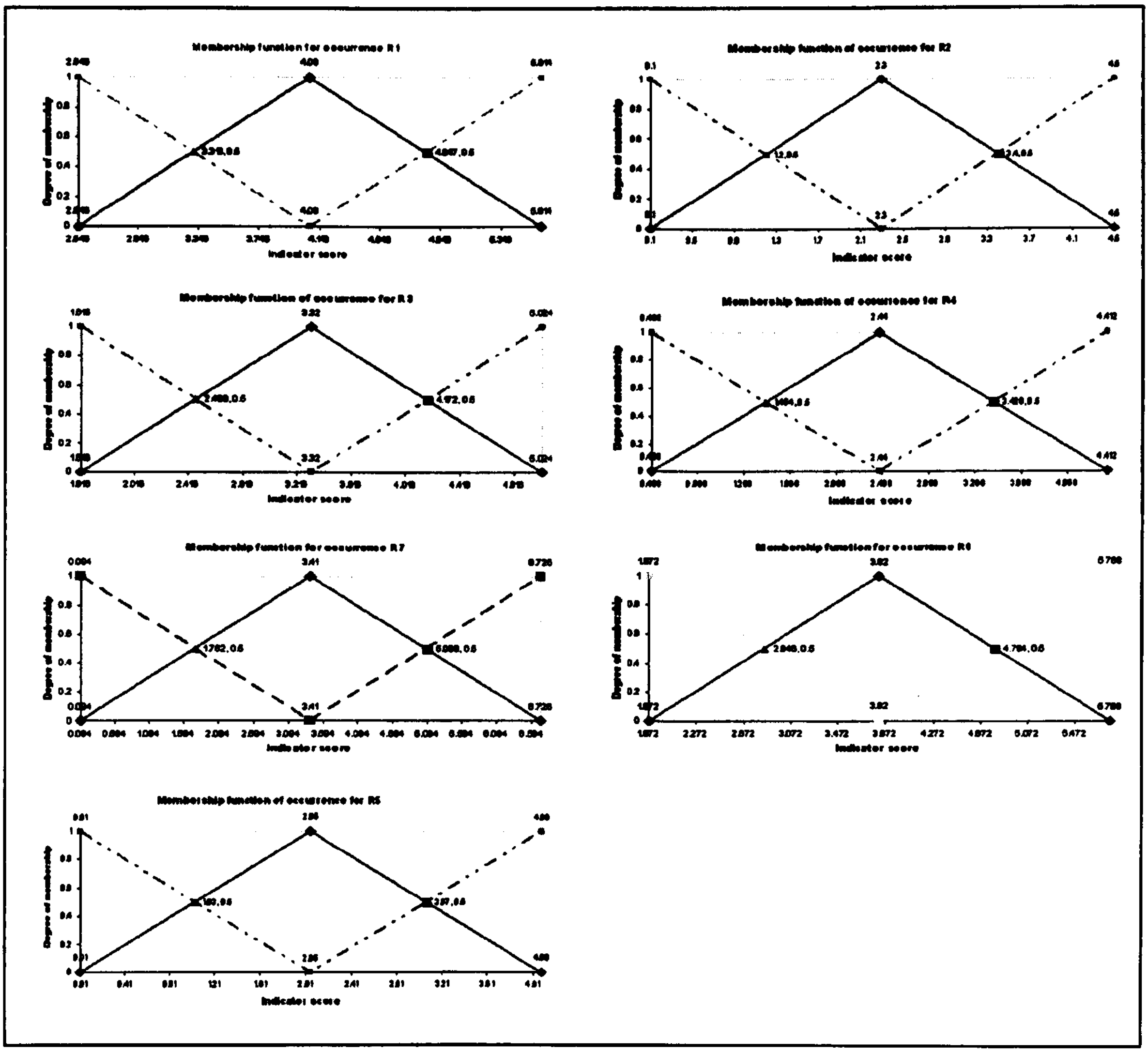


Figure 9.8 : Membership function for Requirement stage of likelihood occurrence

Table 9.4 : Development stage data of the likelihood occurrence

Stage	Risk factor	Average mean, (a)	Standard deviation (b)	a – 2b	a - b	a + b	a + 2b
Development stage	D1: Inproper handover from the requirement team	3.168	1.06	1.048	2.108	4.228	5.288
	D2: Inappropriate development methodology used	4.16	0.854	2.452	3.306	5.014	5.868
	D3: Unsuitable working model and prototype	2.53	0.83	0.87	1.7	3.36	4.19
	D4: Programming language and CASE tool selected not adequate	2.51	1.139	0.232	1.371	3.649	4.788
	D5: High level of technical complexities	3.79	0.944	1.902	2.846	4.734	5.678
	D6: Project involves the use of new technology	2.34	1.036	0.268	1.304	3.376	4.412
	D7: Difficulty in defining the input and output of system	3.127	1.174	0.779	1.953	4.301	5.475
	D8: Immature technology	2.22	0.887	0.446	1.333	3.107	3.994
	D9: Technological advancements and changes	2.06	0.819	0.422	1.241	2.879	3.698
	D10: Failures and inconsistencies of unit/modules test results	2.25	0.975	0.3	1.275	3.225	4.2
	D11: Failure of user acceptance test	3.21	0.937	1.336	2.273	4.147	5.084
	D12: Time consuming for testing	3.69	0.849	1.992	2.841	4.539	5.388
	D13: Resources shifted from project during development due to organisational priorities	3.287	1.034	1.219	2.253	4.321	5.355
	D14: Changes in management of organisation during development	2.02	1.005	0.01	1.015	3.025	4.03
	D15: Lack of users involvement and commitment	3.92	1.139	1.642	2.781	5.059	6.198
	D17: Ineffective communication within development team members	4.3	0.9	2.5	3.4	5.2	6.1
	D21: Inexperienced team members	2.18	1.079	0.022	1.101	3.259	4.338
	D22: Lack of commitment to project among development team members	2.9	1.036	0.828	1.864	3.936	4.972
	D23: Ineffective and inexperienced project manager	4.02	0.721	2.578	3.299	4.741	5.462

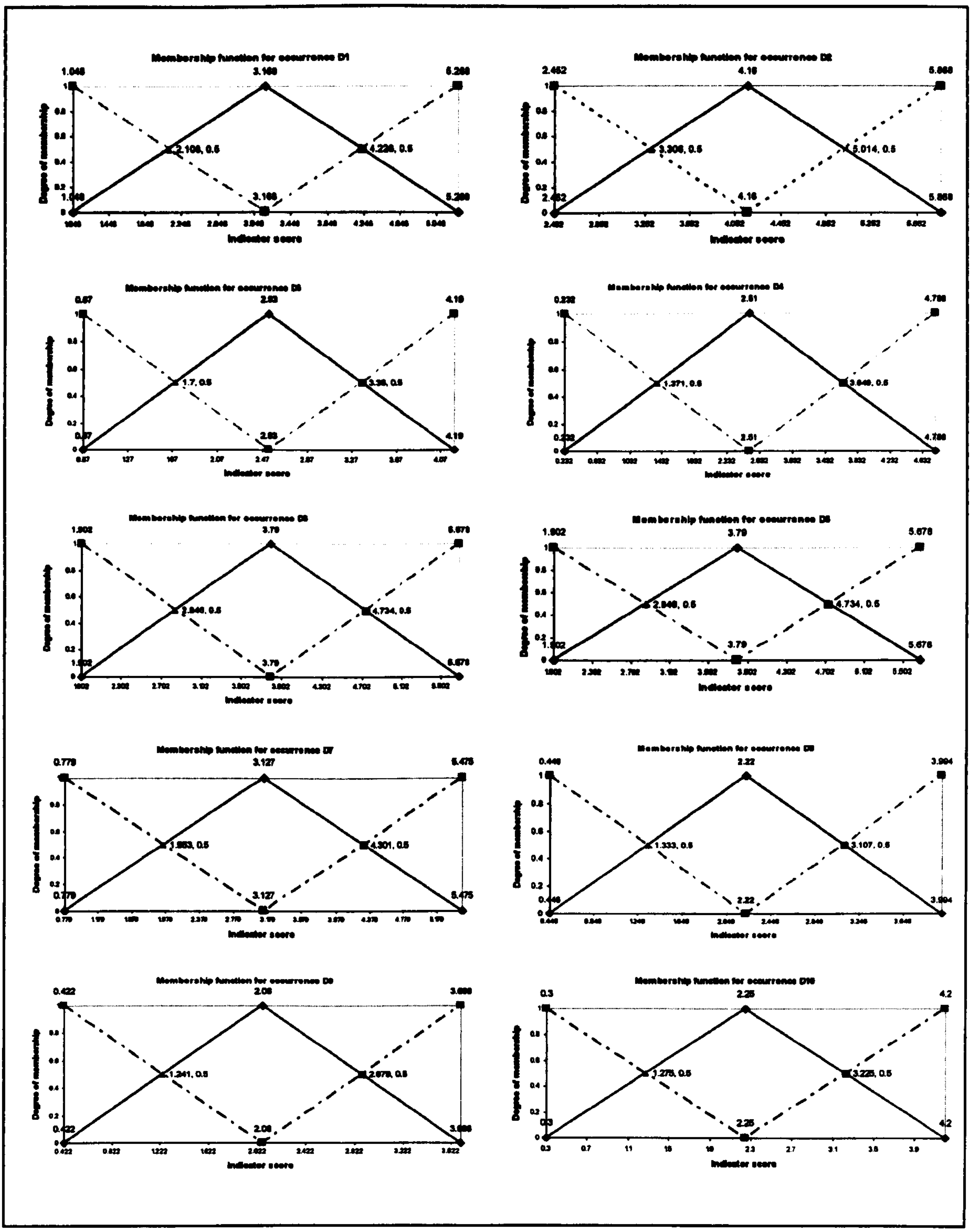
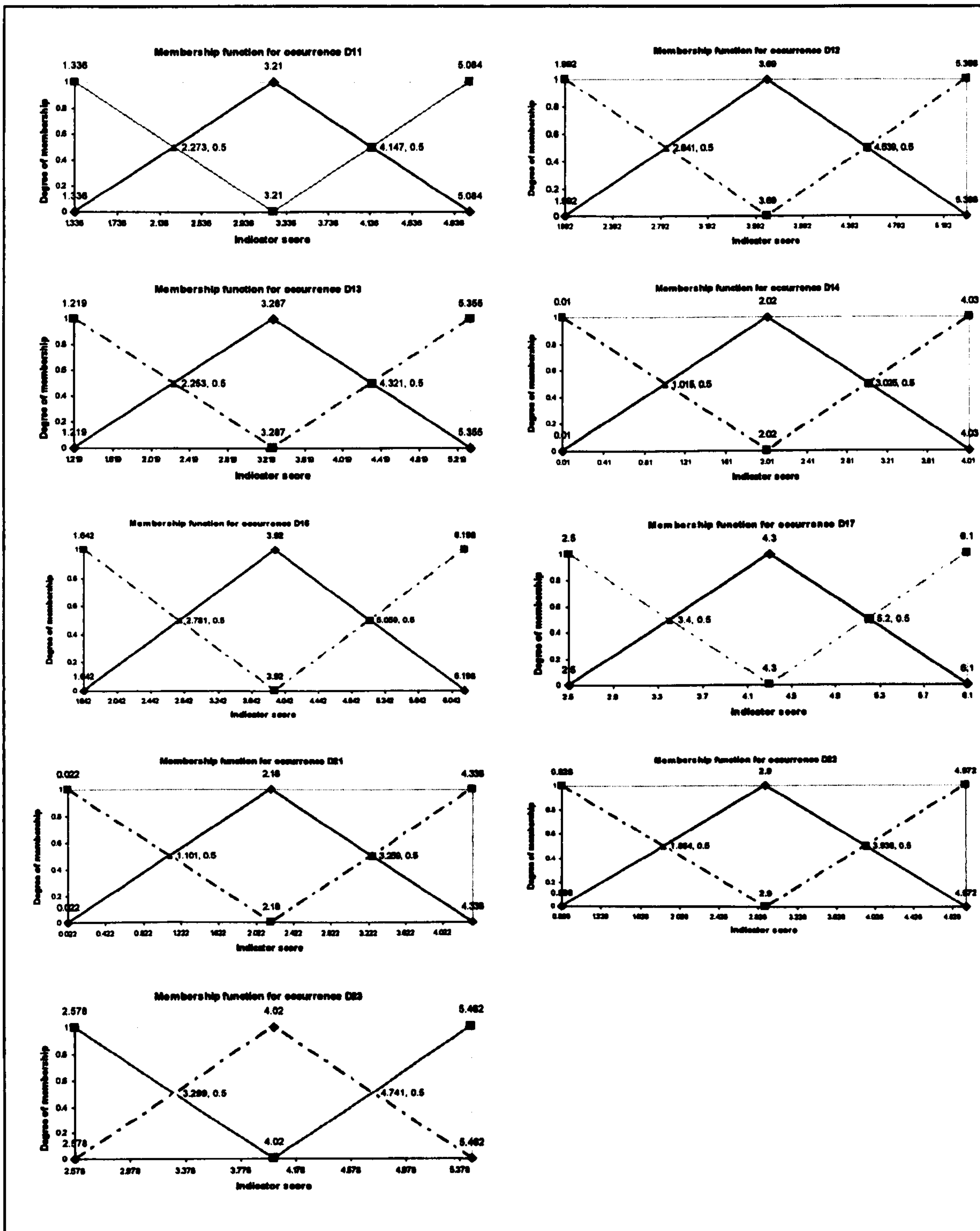


Figure 9.9 : Membership function for Development stage of likelihood occurrence

….membership function for Development stage of likelihood occurrence…. continued



9.8.2. Membership function for Impact of risk factors on cost overrun

The same method and processes applied for the membership function of the risk factors impact on cost overrun, as for the Likelihood occurrence of risk. The mean and the standard deviation is used for developing the MBF. The MBF for various level of significance for each risk factors in the stages was calculated and shown in Table 9.5 – Table 9.8. The resulted figures from Table 9.5 – Table 9.8, were plotted in graphs form to illustrate the MBF of the impact of risk factors on cost overrun for each of the stages. This illustration is shown in Figure 9.10 – Figure 9.13.

Table 9.5 : Feasibility study stage data for impact of risk factor on cost overrun

Stage	Risk factor	Average mean, (a)	Standard deviation, (b)	a – 2b	a - b	a + b	a + 2b
Feasibility study	F1: Improper justification of cost benefit analysis	3.24	1.208	0.824	2.032	4.448	5.656
	F2: Too narrow focus on the technical IT issues	2.58	0.867	0.846	1.713	3.447	4.314
	F3: Overlooked the management and business impact issues	3.44	0.995	1.45	2.445	4.435	5.43
	F4: Wrong justification of investment alternatives and opportunity cost	2.29	0.977	0.336	1.313	3.267	4.244
	F5: Inappropriate technology chosen from the feasibility study	2.02	0.71	0.6	1.31	2.73	3.44

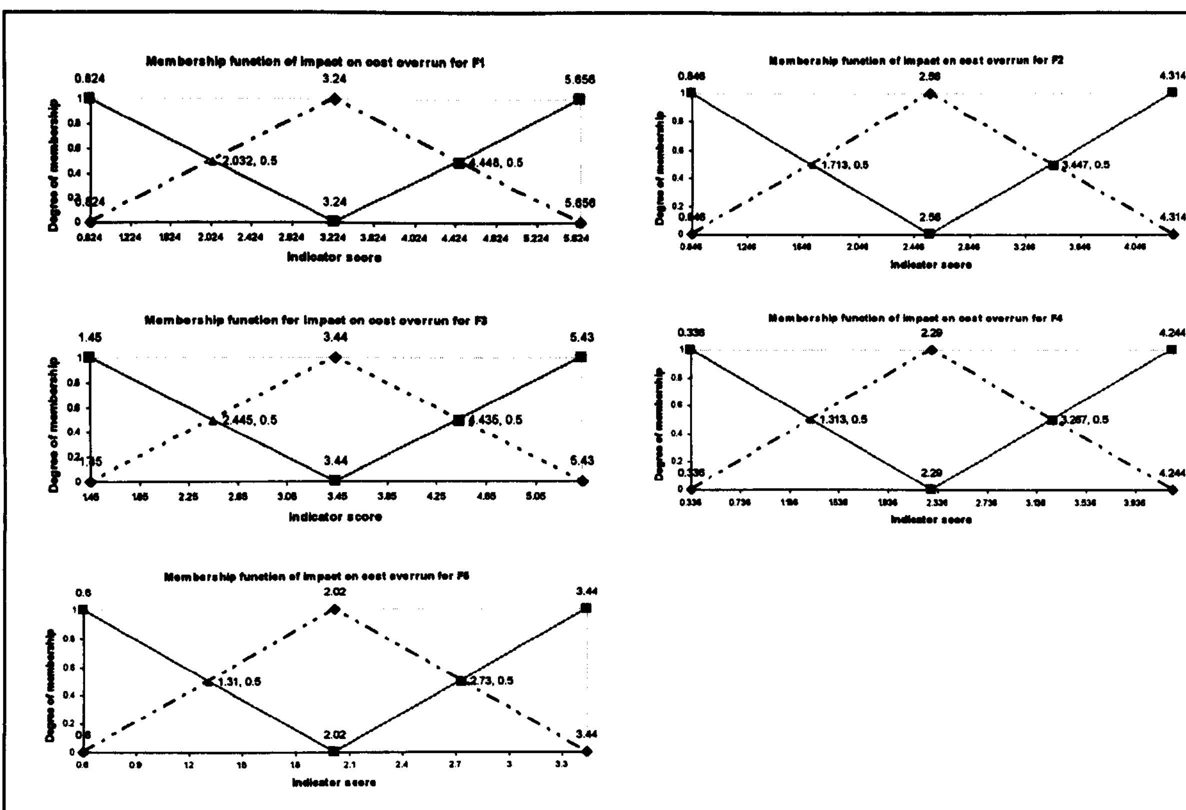


Figure 9.10 : Membership function for Feasibility study stage

Table 9.6 : Project planning stage data for impact of risk factor on cost overrun

Stage	Risk factor	Average mean, (a)	Standard deviation (b)	a – 2b	a - b	a + b	a + 2b
Project Planning stage	P1: Unclear project scope & objectives	4.33	0.572	3.186	3.758	4.902	5.474
	P2: Undefined project success criteria	3.67	0.809	2.052	2.861	4.479	5.288
	P3: Lack of quality control procedure and mechanism	2.56	0.854	0.852	1.706	3.414	4.268
	P4: Project milestones for stages not well established	3.138	1.029	1.08	2.109	4.167	5.196
	P5: Inproper change management plan	3.82	0.818	2.184	3.002	4.638	5.456
	P6: Inaccurate estimate of resources	4.24	0.886	2.468	3.354	5.126	6.012
	P7: Unrealistic project schedule	4.32	0.776	2.768	3.544	5.096	5.872
	P8: Inadequate detail work breakdown structure	2.29	0.978	0.334	1.312	3.268	4.246
	P9: Critical & non-critical activities of project not identified	3.53	0.884	1.762	2.646	4.414	5.298
	P10: Project management & development team not properly set up	1.99	0.978	0.034	1.012	2.968	3.946
	P11: Unclear line of decision making authority throughout project	3.043	1.131	0.781	1.912	4.174	5.305
	P12: Lack on contingency plan/back up	4.21	1.019	2.172	3.191	5.229	6.248
	P13: System conversion method not well planned	2.1	0.783	0.534	1.317	2.883	3.666
	P14: Improper planning of timeframe for project reviews and updating	3.27	1.516	0.238	1.754	4.786	6.302

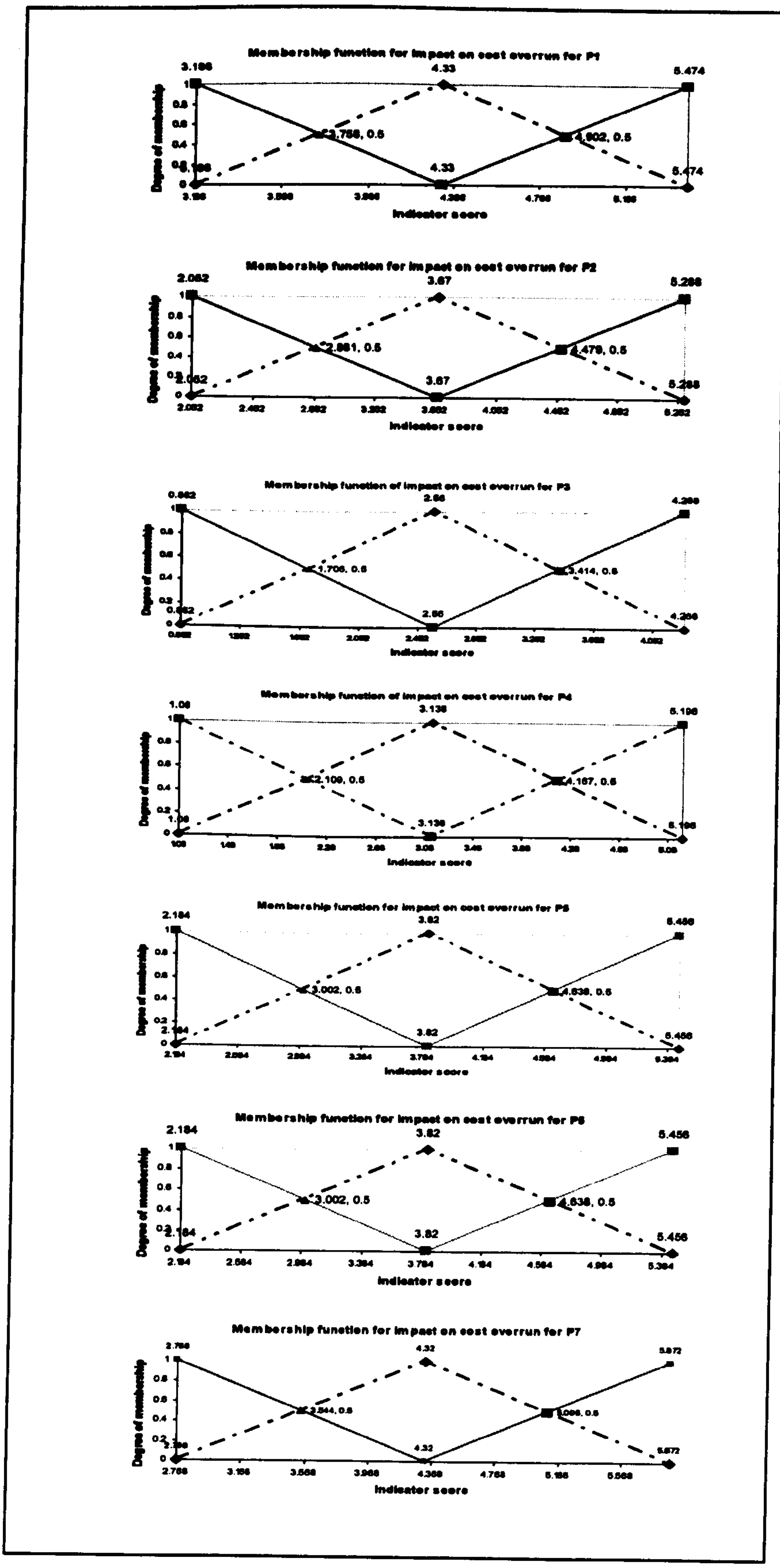


Figure 9.11 : Membership function for project planning stage.

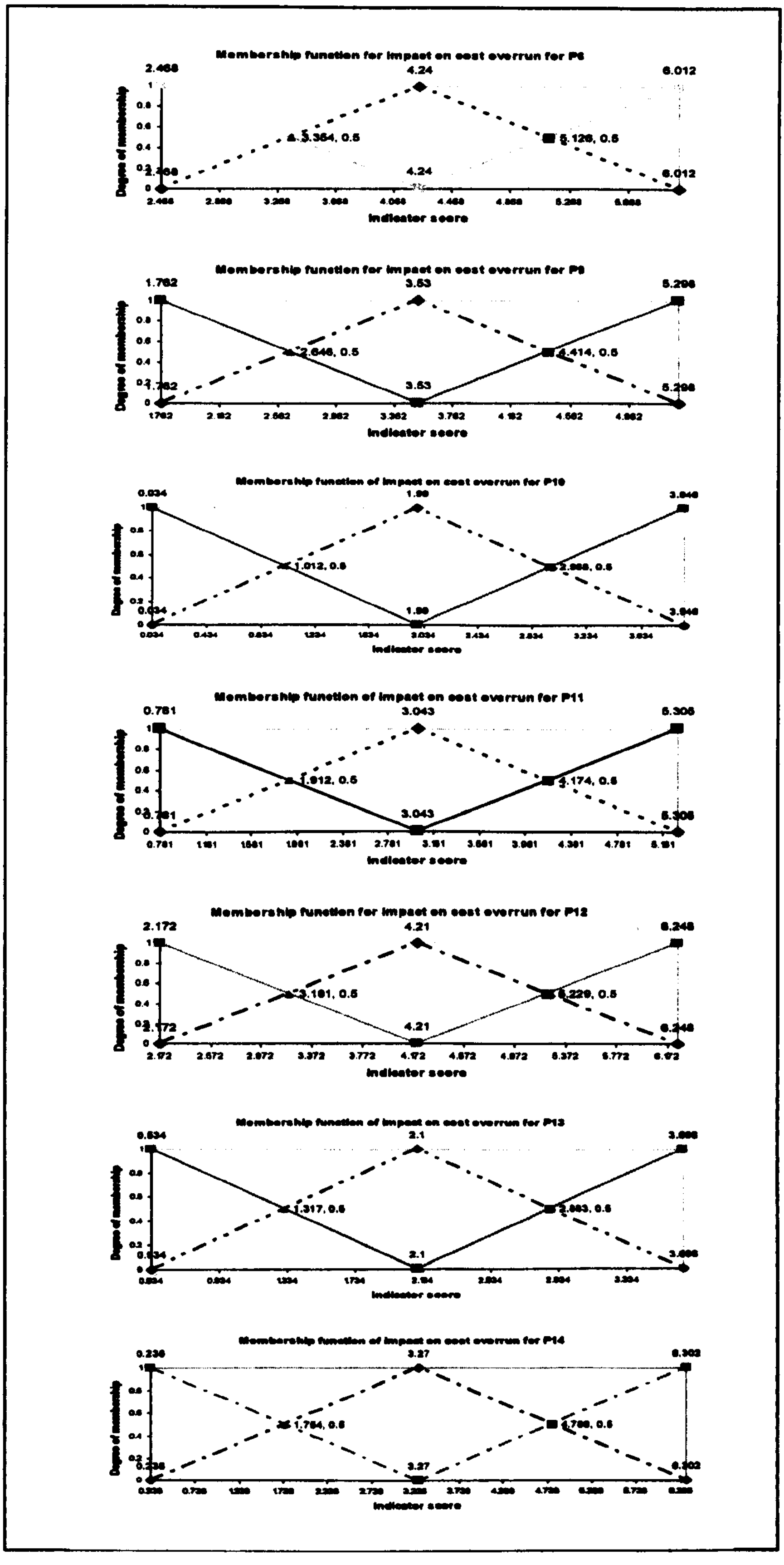


Figure 9.11.... Membership function for planning stage.. cost overrun.. continued

Table 9.7 : Requirement stage data for impact of risk factor on cost overrun

Stage	Risk factor	Average mean, (a)	Standard deviation (b)	a – 2b	a - b	a + b	a + 2b
Requirement stage	R1: Unclear & inadequate identification of systems requirements	3.99	0.867	2.256	3.123	4.857	5.724
	R2: Incorrect systems requirements	2.38	0.98	0.42	1.4	3.36	4.34
	R3: Misinterpretations of systems requirements	3.89	0.827	2.236	3.063	4.717	5.544
	R4: Conflicting system requirements	1.98	0.684	0.612	1.296	2.664	3.348
	R5: Gold plating or unnecessary functions and requirements	1.91	0.899	0.112	1.011	2.809	3.708
	R6: Inadequate validation of systems requirements	3.84	1.456	0.928	2.384	5.296	6.752
	R7: Lack of users involvement in requirement stage	3.241	0.98	1.281	2.261	4.221	5.201

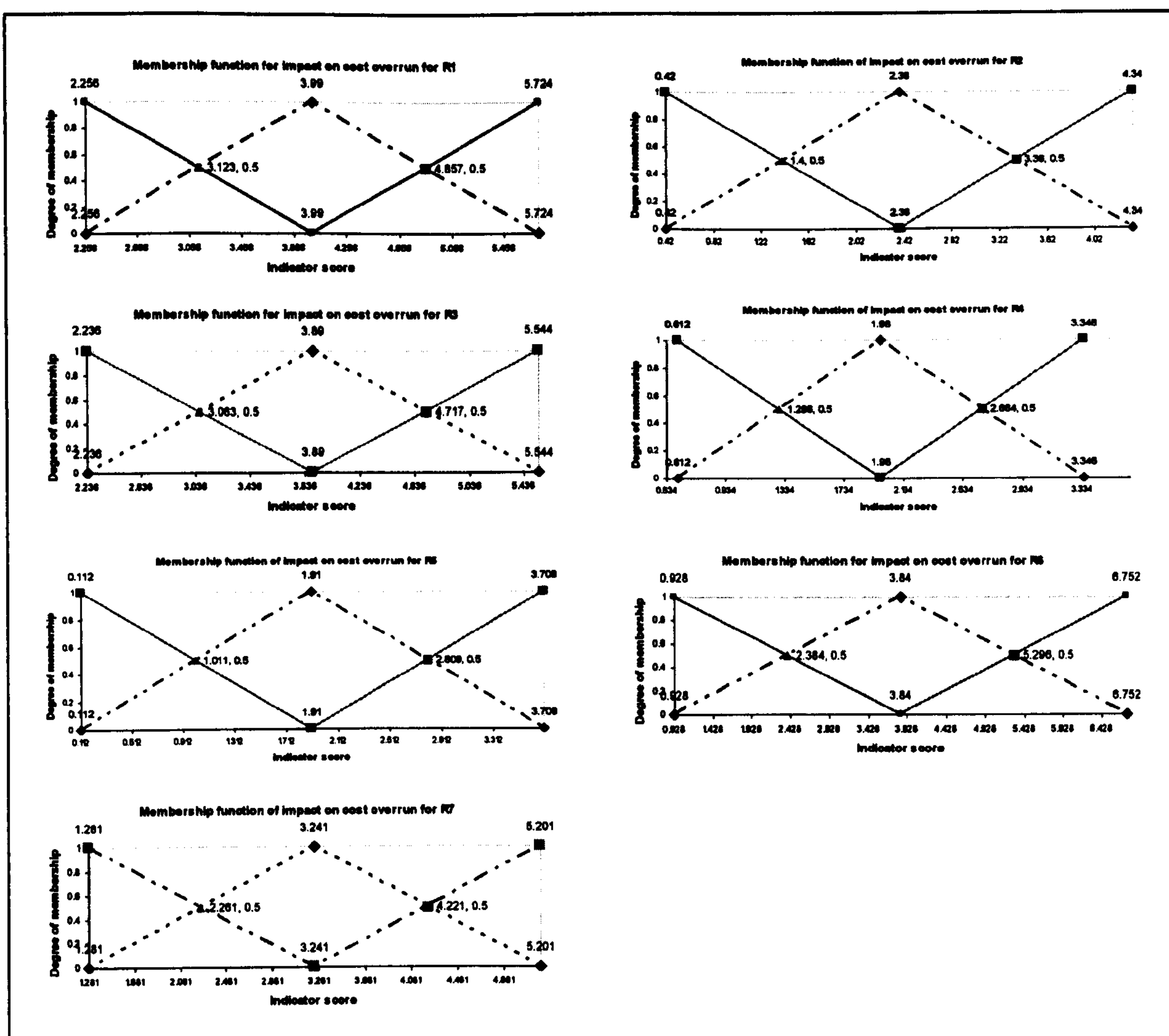


Figure 9.12 : Membership function for requirement stage.

Table 9.8: Development stage data for impact of risk factor on cost overrun

Stage	Risk factor	Average mean, (a)	Standard deviation (b)	a – 2b	a - b	a + b	a + 2b
Development stage	D1: Inproper handover from the requirement team	3.19	1.366	0.458	1.824	4.556	5.922
	D2: Inappropriate development methodology used	3.62	1.102	1.416	2.518	4.722	5.824
	D3: Unsuitable working model and prototype	2.29	0.983	0.324	1.307	3.273	4.256
	D4: Programming language and CASE tool selected not adequate	2.54	1.133	0.274	1.407	3.673	4.806
	D5: High level of technical complexities	3.28	1.113	1.054	2.167	4.393	5.506
	D6: Project involves the use of new technology	2.67	0.973	0.724	1.697	3.643	4.616
	D7: Difficulty in defining the input and output of system	2.31	0.968	0.374	1.342	3.278	4.246
	D8: Immature technology	2.06	0.825	0.41	1.235	2.885	3.71
	D9: Technological advancements and changes	2.16	1.02	0.12	1.14	3.18	4.2
	D10: Failures and inconsistencies of unit/modules test results	2.29	0.983	0.324	1.307	3.273	4.256
	D11: Failure of user acceptance test	4.0	1.452	1.096	2.548	5.452	6.904
	D12: Time consuming for testing	2.43	0.854	0.722	1.576	3.284	4.138
	D13: Resources shifted from project during development due to organisational priorities	3.141	1.056	1.029	2.085	4.197	5.253
	D14: Changes in management of organisation during development	2.14	1.068	0.004	1.072	3.208	4.276
	D15: Lack of users involvement and commitment	3.90	1.182	1.536	2.718	5.082	6.264
	D17: Ineffective communication within development team members	3.12	1.458	0.204	1.662	4.578	6.036
	D19: Inadequately trained development team members	1.95	0.751	0.448	1.199	2.701	3.452
	D22: Lack of commitment to project among development team members	2.55	1.116	0.318	1.434	3.666	4.782
	D23: Ineffective and inexperienced project manager	3.94	0.766	2.408	3.174	4.706	5.472

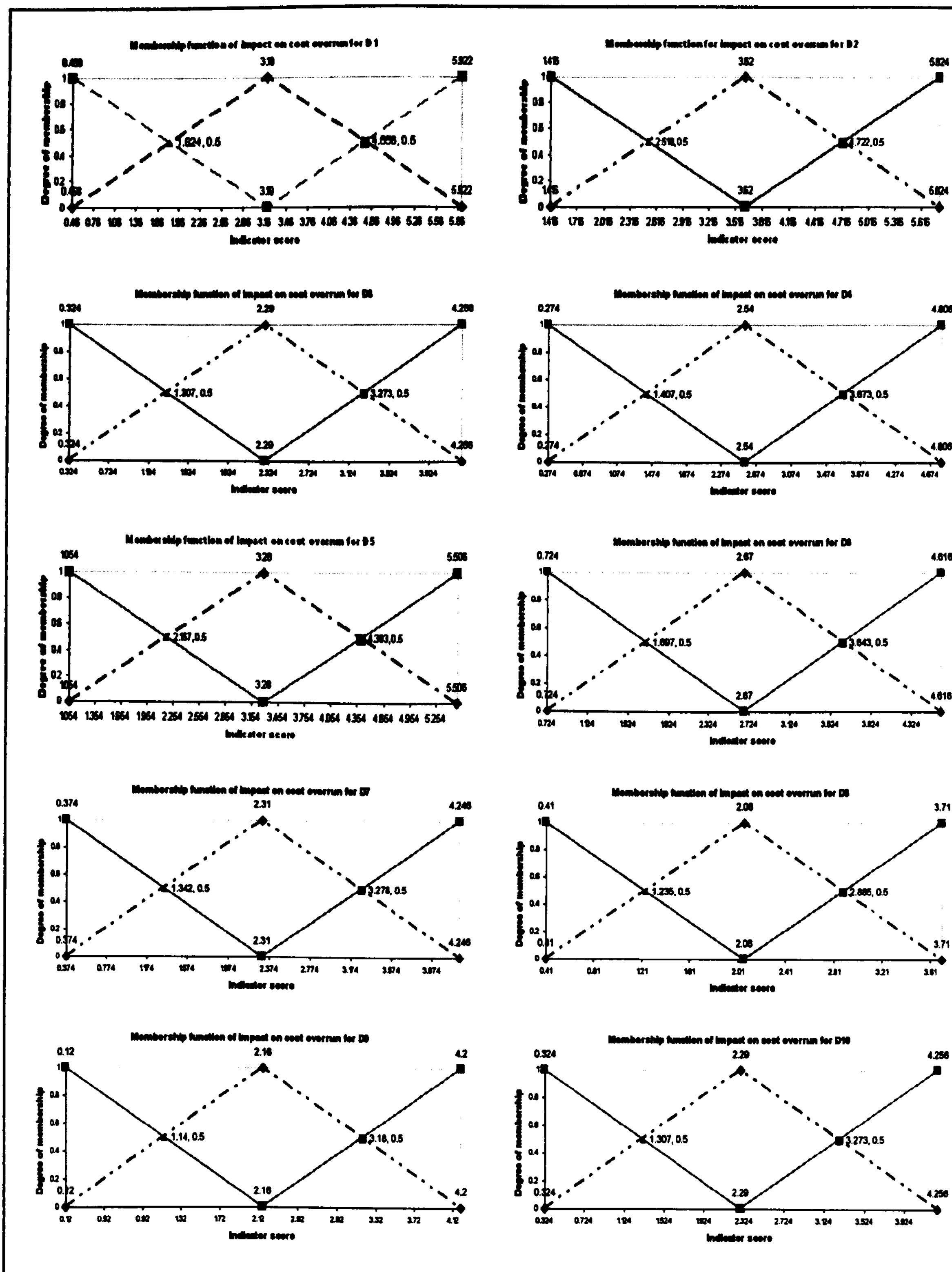


Figure 9.13a : Membership function for development stage.

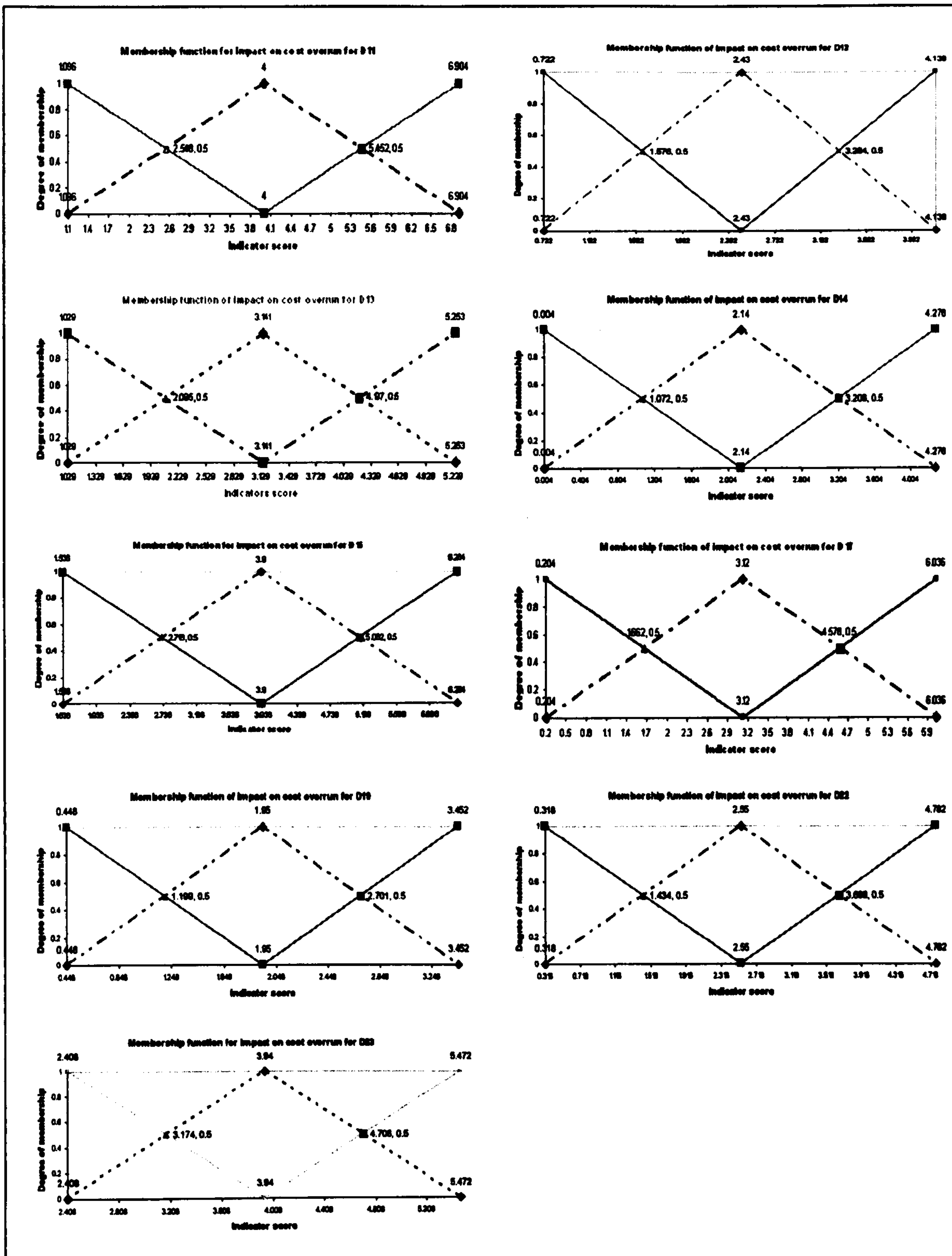


Figure 9.13b : Membership function for development stage.

9.9. Fuzzy computation combinations

Figure 9.14 and Table 9.8, presents the possible fuzzy combinations and scenarios for each risk factor. Each of these combinations may have several other alternatives depending on the degree of membership and levels of Alpha-cuts of the selected combination. The MBF for each risk factor chosen will then be calculated based on the possible combinations and weights score from the fuzzy computation. Boussabaine and Elhag stated that, eventhough weight can be expressed in either numeric (crisp) or linguistics (fuzzy) terms, but all the weights must be defined in the same manner. As a result of this, the fuzzy weighted average used for this study is extracted using the linguistic weights.

Hence, linguistic weights are being expressed in terms of degree of significance of risk factors, which include, Low, Moderate and High. The combination of MBF of risk factors and MBF of degree of significance is used to develop the model. As the possibilities of occurrence for both MBF of risk factors and MBF of degree of significance is 3 each, the possibilities of combinations for each risk factor for this study could be 9 (which is 3 x 3).

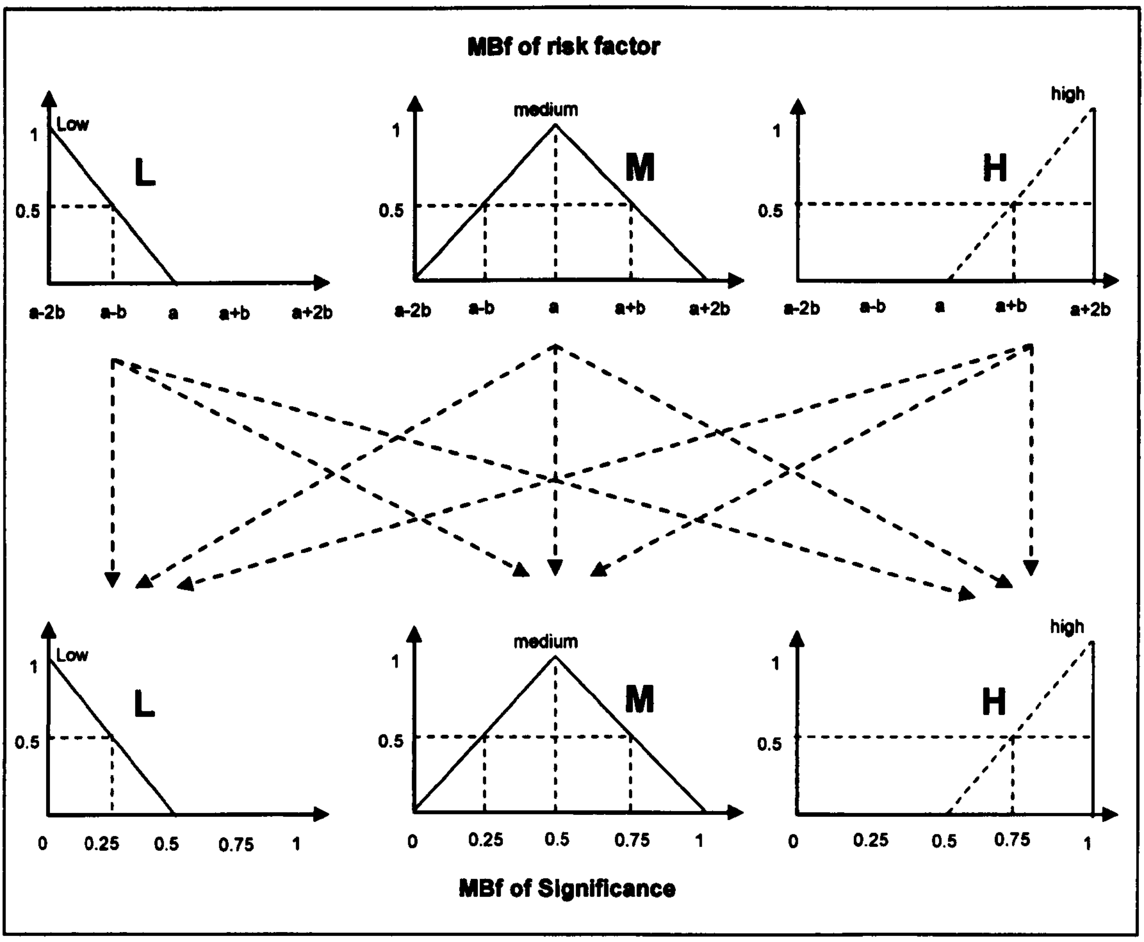


Figure 9.14 : Number of possible fuzzy computation combinations.

Table 9.9 : Possible combinations

	Low risk occurrence (L)	Medium risk occurrence (M)	High risk occurrence (H)
Low Significance (L)	LL	ML	HL
Medium Sognificance (M)	LM	MM	HM
High Significance (H)	LH	MH	HH

Using the Formula B, examples of the fuzzy computation for the combination for each of the risk factor for the stages in shown in Table 9.9 -Table 9.12 (for likelihood occurrence) and Table 9.13 – Table 9.16 (for impact of risk factors on the cost overrun).

Likelihood occurrence :

Table 9.10 – 9.13 : Example of Possible alternatives of Fuzzy computation of the category of responses for the likelihood occurrence of risks factors

Table 9.10 : Fuzzy computation for Feasibility stage of likelihood occurrence

alpha cuts	Low			Medium			High		
	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX
F1									
alpha cut : 0	0.3	3.02	0.906	0.4	4.72	1.888	0.8	5.12	4.096
alphs cut : 0.5	0.4	2.85	1.14	0.5	4.54	2.27	0.7	4.92	3.444
alpha cut : 1	0.5	2.66	1.33	0.8	4	3.2	0.6	4.72	2.832
	1.2	Sum WX	**3.376**	**1.7**	Sum WX	**7.358**	**2.1**	Sum WX	**10.372**
F2									
alpha cut : 0	0.2	2.87	0.574	0.4	4.27	1.708	0.6	4.3	2.58
alphs cut : 0.5	0.3	2.67	0.801	0.4	4.27	1.708	0.7	4.5	3.15
alpha cut : 1	0.4	2.5	1	0.7	3.8	2.66	0.7	4.5	3.15
	0.9	Sum WX	**2.375**	**1.5**	Sum WX	**6.076**	**2**	Sum WX	**8.88**
F3									
alpha cut : 0	0.3	2.8	0.84	0.4	4.75	1.9	0.8	5.15	4.12
alphs cut : 0.5	0.4	2.6	1.04	0.5	4.53	2.265	0.7	4.95	3.465
alpha cut : 1	0.5	2.41	1.205	0.8	3.9	3.12	0.6	4.75	2.85
	1.2	Sum WX	**3.085**	**1.7**	Sum WX	**7.285**	**2.1**	Sum WX	**10.435**
F4									
alpha cut : 0	0.2	1.91	0.382	0.3	3.45	1.035	0.7	3.45	2.415
alphs cut : 0.5	0.3	1.75	0.525	0.5	3.115	1.5575	0.6	3.25	1.95
alpha cut : 1	0.4	1.61	0.644	0.6	2.95	1.77	0.5	3.115	1.5575
	0.9	Sum WX	**1.551**	**1.4**	Sum WX	**4.3625**	**1.8**	Sum WX	**5.9225**
F5									
alpha cut : 0	0.2	2.1	0.42	0.3	3.65	1.095	0.7	3.65	2.555
alphs cut : 0.5	0.3	1.9	0.57	0.5	3.302	1.651	0.6	3.5	2.1
alpha cut : 1	0.4	1.72	0.688	0.6	3.2	1.92	0.5	3.302	1.651
	0.9	Sum WX	**1.678**	**1.4**	Sum WX	**4.666**	**1.8**	Sum WX	**6.306**

	Low	Medium	High
	Sum WX = 12.065 **Sum W = 5.1** **Sum WX / Sum W = 2.37**	**Sum WX = 29.748** **Sum W = 7.7** **Sum WX / Sum W = 3.86**	**Sum WX = 41.916** **Sum W = 9.8** **Sum WX / Sum W = 4.28**

Table 9.11 : Fuzzy computation for Project planning stage of likelihood occurrence

alpha cuts	**Low**			**Medium**			**High**		
	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX
P1									
alpha cut : 0	0.2	3.73	0.746	0.5	5.24	2.62	0.8	5.9	4.72
alphs cut : 0.5	0.3	3.53	1.059	0.8	4.6	3.68	0.8	5.9	4.72
alpha cut : 1	0.7	2.65	1.855	0.9	4.4	3.96	0.6	5.45	3.27
	1.2	Sum WX	**3.66**	**2.2**	Sum WX	**10.26**	**2.2**	Sum WX	**12.71**
P2									
alpha cut : 0	0.2	3.35	0.67	0.4	4.7	1.88	0.8	5.03	4.024
alphs cut : 0.5	0.3	3.2	0.96	0.7	4.2	2.94	0.7	4.85	3.395
alpha cut : 1	0.6	2.7	1.62	0.9	3.85	3.465	0.6	4.7	2.82
	1.1	Sum WX	**3.25**	**2**	Sum WX	**8.285**	**2.1**	Sum WX	**10.239**
P3									
alpha cut : 0	0.2	2.3	0.46	0.4	3.8	1.52	0.7	4.1	2.87
alphs cut : 0.5	0.2	2.3	0.46	0.5	3.684	1.842	0.5	3.684	1.842
alpha cut : 1	0.6	1.59	0.954	0.9	2.85	2.565	0.5	3.684	1.842
	1	Sum WX	**1.874**	**1.8**	Sum WX	**5.927**	**1.7**	Sum WX	**6.554**
P4									
alpha cut : 0	0.2	2.1	0.42	0.4	3.4	1.36	0.6	3.4	2.04
alphs cut : 0.5	0.3	1.85	0.555	0.6	3.1	1.86	0.6	3.4	2.04
alpha cut : 1	0.6	1.4	0.84	0.9	2.55	2.295	0.5	3.239	1.6195
	1.1	Sum WX	**1.815**	**1.9**	Sum WX	**5.515**	**1.7**	Sum WX	**5.6995**
P5									
alpha cut : 0	0.2	3.2	0.64	0.4	4.8	1.92	0.7	4.95	3.465
alphs cut : 0.5	0.3	3	0.9	0.6	4.4	2.64	0.6	4.8	2.88
alpha cut : 1	0.7	2.1	1.47	0.9	3.8	3.42	0.5	4.536	2.268
	1.2	Sum WX	**3.01**	**1.9**	Sum WX	**7.98**	**1.8**	Sum WX	**8.613**
P6									
alpha cut : 0	0.2	4.1	0.82	0.5	3.586	1.793	0.8	5.8	4.64
alphs cut : 0.5	0.3	3.93	1.179	0.7	4.93	3.451	0.7	5.63	3.941
alpha cut : 1	0.7	3.23	2.261	0.9	4.6	4.14	0.7	5.63	3.941
	1.2	Sum WX	**4.26**	**2.1**	Sum WX	**9.384**	**2.2**	Sum WX	**12.522**
P7									
alpha cut : 0	0.2	4.1	0.82	0.5	3.626	1.813	0.8	5.65	4.52
alphs cut : 0.5	0.3	3.94	1.182	0.7	4.9	3.43	0.8	5.65	4.52
alpha cut : 1	0.7	3.3	2.31	0.9	4.5	4.05	0.6	5.34	3.204
	1.2	Sum WX	**4.312**	**2.1**	Sum WX	**9.293**	**2.2**	Sum WX	**12.244**
P8									
alpha cut : 0	0.2	2.18	0.436	0.3	3.85	1.155	0.7	3.98	2.786
alphs cut : 0.5	0.2	2.18	0.436	0.6	3.38	2.028	0.5	3.546	1.773
alpha cut : 1	0.6	1.38	0.828	0.9	2.75	2.475	0.6	3.75	2.25
	1	Sum WX	**1.7**	**1.8**	Sum WX	**5.658**	**1.8**	Sum WX	**6.809**

P9									
alpha cut : 0	0.2	3.3	0.66	0.4	4.95	1.98	0.8	5.35	4.28
alphs cut : 0.5	0.3	3.1	0.93	0.7	4.35	3.045	0.7	5.15	3.605
alpha cut : 1	0.6	2.5	1.5	0.9	3.95	3.555	0.6	4.95	2.97
	1.1	Sum WX	**3.09**	**2**	Sum WX	**8.58**	**2.1**	Sum WX	**10.855**
P10									
alpha cut : 0	0.2	1.55	0.31	0.4	3.02	1.208	0.7	3.22	2.254
alphs cut : 0.5	0.2	1.55	0.31	0.6	2.62	1.572	0.6	3.02	1.812
alpha cut : 1	0.6	0.8	0.48	0.9	2.05	1.845	0.6	3.02	1.812
	1	Sum WX	**1.1**	**1.9**	Sum WX	**4.625**	**1.9**	Sum WX	**5.878**
P11									
alpha cut : 0	0.2	3.6	0.72	0.5	4.88	2.44	0.8	5.4	4.32
alphs cut : 0.5	0.3	3.45	1.035	0.7	4.5	3.15	0.7	4.5	3.15
alpha cut : 1	0.6	2.85	1.71	0.9	4.15	3.735	0.6	5.05	3.03
	1.1	Sum WX	**3.465**	**2.1**	Sum WX	**9.325**	**2.1**	Sum WX	**10.5**
P12									
alpha cut : 0	0.2	3.4	0.68	0.5	4.275	2.1375	0.8	4.65	3.72
alphs cut : 0.5	0.3	3.3	0.99	0.7	4.03	2.821	0.7	4.5	3.15
alpha cut : 1	0.7	2.8	1.96	0.8	3.9	3.12	0.7	4.5	3.15
	1.2	Sum WX	**3.63**	**2**	Sum WX	**8.0785**	**2.2**	Sum WX	**10.02**
P13									
alpha cut : 0	0.2	3	0.6	0.5	4.265	2.1325	0.7	4.645	3.2515
alphs cut : 0.5	0.3	2.75	0.825	0.6	4.045	2.427	0.7	4.645	3.2515
alpha cut : 1	0.7	1.95	1.365	0.8	3.65	2.92	0.6	4.445	2.667
	1.2	Sum WX	**2.79**	**1.9**	Sum WX	**7.4795**	**2**	Sum WX	**9.17**
P14									
alpha cut : 0	0.2	2.85	0.57	0.5	4.01	2.005	0.8	4.45	3.56
alphs cut : 0.5	0.3	2.65	0.795	0.5	4.01	2.005	0.7	4.35	3.045
alpha cut : 1	0.7	2.050	1.435	0.8	3.5	2.8	0.6	4.15	2.49
	1.2	Sum WX	**2.8**	**1.8**	Sum WX	**6.81**	**2.1**	Sum WX	**9.095**

Sum WX = 40.756 **Sum W = 15.8** **Sum WX / Sum W = 2.58**	**Sum WX = 107.2** **Sum W = 27.5** **Sum WX / Sum W = 3.9**	**Sum WX = 130.908** **Sum W = 28.1** **Sum WX / Sum W = 4.66**

Table 9.12 : Fuzzy computation for Requirement stage of likelihood occurrence

alpha cuts	Low			Medium			High		
	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX
R1									
alpha cut : 0	0.2	3.75	0.75	0.4	5	2	0.8	5.3	4.24
alphs cut : 0.5	0.3	3.6	1.08	0.7	4.55	3.185	0.7	5.15	3.605
alpha cut : 1	0.5	3.313	1.6565	0.9	4.2	3.78	0.7	5.15	3.605
	1	Sum WX	**3.4865**	**2**	Sum WX	**8.965**	**2.2**	Sum WX	**11.45**
R2									
alpha cut : 0	0.2	1.85	0.37	0.3	3.85	1.155	0.6	3.6	2.16
alphs cut : 0.5	0.3	1.6	0.48	0.4	3.6	1.44	0.5	3.4	1.7
alpha cut : 1	0.4	1.4	0.56	0.7	2.9	2.03	0.6	3.6	2.16
	0.9	Sum WX	**1.41**	**1.4**	Sum WX	**4.625**	**1.7**	Sum WX	**6.02**
R3									
alpha cut : 0	0.2	3	0.6	0.3	4.5	1.35	0.6	4.35	2.61
alphs cut : 0.5	0.3	2.8	0.84	0.7	3.8	2.66	0.5	4.172	2.086
alpha cut : 1	0.4	2.6	1.04	0.8	3.65	2.92	0.5	4.172	2.086
	0.9	Sum WX	**2.48**	**1.8**	Sum WX	**6.93**	**1.6**	Sum WX	**6.782**
R4									
alpha cut : 0	0.2	2	0.4	0.3	3.75	1.125	0.5	3.426	1.713
alphs cut : 0.5	0.3	1.75	0.525	0.4	3.6	1.44	0.5	3.426	1.713
alpha cut : 1	0.3	1.75	0.525	0.6	3.2	1.92	0.5	3.436	1.718
	0.8	Sum WX	**1.45**	**1.3**	Sum WX	**4.485**	**1.5**	Sum WX	**5.144**
R5									
alpha cut : 0	0.2	1.6	0.32	0.3	3.4	1.02	0.6	3.25	1.95
alphs cut : 0.5	0.2	1.6	0.32	0.5	3.07	1.535	0.6	3.25	1.95
alpha cut : 1	0.4	1.2	0.48	0.6	2.9	1.74	0.5	3.07	1.535
	0.8	Sum WX	**1.12**	**1.4**	Sum WX	**4.295**	**1.7**	Sum WX	**5.435**
R6									
alpha cut : 0	0.2	3.4	0.68	0.4	4.95	1.98	0.8	5.35	4.28
alphs cut : 0.5	0.3	3.2	0.96	0.6	4.6	2.76	0.7	5.15	3.605
alpha cut : 1	0.6	2.6	1.56	0.9	3.95	3.555	0.7	5.15	3.605
	1.1	Sum WX	**3.2**	**1.9**	Sum WX	**8.295**	**2.2**	Sum WX	**11.49**
R7									
alpha cut : 0	0.2	2.7	0.54	0.4	5.35	2.14	0.7	5.7	3.99
alphs cut : 0.5	0.4	2.09	0.836	0.5	5.07	2.535	0.6	5.35	3.21
alpha cut : 1	0.5	1.752	0.876	0.8	4.09	3.272	0.6	5.35	3.21
	1.1	Sum WX	**2.252**	**1.7**	Sum WX	**7.947**	**1.9**	Sum WX	**10.41**
	Sum WX = 15.399 **Sum W = 6.6** **Sum WX / Sum W = 2.33**			**Sum WX = 45.542** **Sum W = 11.5** **Sum WX / Sum W = 3.96**			**Sum WX = 56.731** **Sum W = 12.8** **Sum WX / Sum W = 4.43**		

Table 9.13 : Fuzzy computation for Development stage of likelihood occurrence

alpha cuts	Low			Medium			High		
	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX
D1									
alpha cut : 0	0.2	2.7	0.54	0.3	1.048	0.3144	0.5	4.23	2.115
alpha cut : 0.5	0.2	2.7	0.54	0.4	4.65	1.86	0.5	4.23	2.115
alpha cut : 1	0.5	2.108	1.054	0.7	3.8	2.66	0.4	4.45	1.78
	0.9	Sum WX	**2.134**	**1.4**	Sum WX	**4.8344**	**1.4**	Sum WX	**6.01**
D2									
alpha cut : 0	0.2	3.8	0.76	0.4	5.15	2.06	0.8	5.55	4.44
alpha cut : 0.5	0.3	3.65	1.095	0.8	4.5	3.6	0.8	5.55	4.44
alpha cut : 1	0.5	3.306	1.653	0.9	4.3	3.87	0.7	4.65	3.255
	1	Sum WX	**3.508**	**2.1**	Sum WX	**9.53**	**2.3**	Sum WX	**12.135**
D3									
alpha cut : 0	0.2	2.15	0.43	0.3	3.67	1.101	0.5	3.36	1.68
alpha cut : 0.5	0.2	2.15	0.43	0.4	3.5	1.4	0.5	3.36	1.68
alpha cut : 1	0.5	1.700	0.85	0.7	2.95	2.065	0.5	3.36	1.68
	0.9	Sum WX	**1.71**	**1.4**	Sum WX	**4.566**	**1.5**	Sum WX	**5.04**
D4									
alpha cut : 0	0.2	2.03	0.406	0.4	3.83	1.532	0.7	4.15	2.905
alpha cut : 0.5	0.3	1.8	0.54	0.8	3	2.4	0.7	4.15	2.905
alpha cut : 1	0.5	1.371	0.6855	0.9	2.75	2.475	0.7	4.15	2.905
	1	Sum WX	**1.6315**	**2.1**	Sum WX	**6.407**	**2.1**	Sum WX	**8.715**
D5									
alpha cut : 0	0.5	2.846	1.423	0.7	4.45	3.115	0.8	5.3	4.24
alpha cut : 0.5	0.7	2.45	1.715	0.8	4.15	3.32	0.8	5.3	4.24
alpha cut : 1	0.7	2.45	1.715	0.9	4	3.6	0.8	5.3	4.24
	1.9	Sum WX	**4.853**	**2.4**	Sum WX	**10.035**	**2.4**	Sum WX	**12.72**
D6									
alpha cut : 0	0.2	1.9	0.38	0.4	3.55	1.42	0.7	2.95	2.065
alpha cut : 0.5	0.3	1.7	0.51	0.8	2.75	2.2	0.8	2.75	2.2
alpha cut : 1	0.5	1.304	0.652	0.9	2.55	2.295	0.7	2.95	2.065
	1	Sum WX	**1.542**	**2.1**	Sum WX	**5.915**	**2.2**	Sum WX	**6.33**
D7									
alpha cut : 0	0.3	2.37	0.711	0.4	4.5	1.8	0.5	4.301	2.1505
alpha cut : 0.5	0.4	2.17	0.868	0.5	4.301	2.1505	0.6	4.58	2.748
alpha cut : 1	0.5	1.953	0.9765	0.6	4.05	2.43	0.6	4.58	2.748
	1.2	Sum WX	**2.5555**	**1.5**	Sum WX	**6.3805**	**1.7**	Sum WX	**7.6465**
D8									
alpha cut : 0	0.2	1.846	0.3692	0.4	3.25	1.3	0.8	3.646	2.9168
alpha cut : 0.5	0.3	1.646	0.4938	0.5	3.107	1.5535	0.7	3.446	2.4122
alpha cut : 1	0.5	1.333	0.6665	0.7	2.75	1.925	0.7	3.446	2.4122
	1	Sum WX	**1.5295**	**1.6**	Sum WX	**4.7785**	**2.2**	Sum WX	**7.7412**
D9									
alpha cut : 0	0.2	1.75	0.35	0.3	3.2	0.96	0.5	2.879	1.4395
alpha cut : 0.5	0.2	1.75	0.35	0.4	3.02	1.208	0.5	2.879	1.4395
alpha cut : 1	0.5	1.241	0.6205	0.7	2.55	1.785	0.4	2.7	1.08
	0.9	Sum WX	**1.3205**	**1.4**	Sum WX	**3.953**	**1.4**	Sum WX	**3.959**

D10									
alpha cut : 0	0.2	1.85	0.37	0.4	3.4	1.36	0.6	3.4	2.04
alpha cut : 0.5	0.3	1.65	0.495	0.5	3.225	1.6125	0.8	3.8	3.04
alpha cut : 1	0.5	1.275	0.6375	0.7	2.8	1.96	0.7	3.6	2.52
	1	Sum WX	1.5025	1.6	Sum WX	4.9325	2.1	Sum WX	7.6
D11									
alpha cut : 0	0.4	2.5	1	0.5	4.15	2.075	0.6	4.35	2.61
alpha cut : 0.5	0.5	2.27	1.135	0.6	3.95	2.37	0.7	4.5	3.15
alpha cut : 1	0.6	2.1	1.26	0.7	3.75	2.625	0.7	4.5	3.15
	1.5	Sum WX	3.395	1.8	Sum WX	7.07	2	Sum WX	8.91
D12									
alpha cut : 0	0.4	2.99	1.196	0.6	4.35	2.61	0.8	5	4
alpha cut : 0.5	0.6	2.65	1.59	0.8	3.99	3.192	0.8	5	4
alpha cut : 1	0.7	2.45	1.715	0.9	3.8	3.42	0.8	5	4
	1.7	Sum WX	4.501	2.3	Sum WX	9.222	2.4	Sum WX	12
D13									
alpha cut : 0	0.2	2.85	0.57	0.4	4.55	1.82	0.6	4.55	2.73
alpha cut : 0.5	0.3	2.65	0.795	0.5	4.321	2.1605	0.6	4.55	2.73
alpha cut : 1	0.6	2.02	1.212	0.8	3.7	2.96	0.6	4.55	2.73
	1.1	Sum WX	2.577	1.7	Sum WX	6.9405	1.8	Sum WX	8.19
D14									
alpha cut : 0	0.2	1.61	0.322	0.3	3.41	1.023	0.5	3.025	1.5125
alpha cut : 0.5	0.3	1.35	0.405	0.5	3.025	1.5125	0.6	3.21	1.926
alpha cut : 1	0.5	1.015	0.5075	0.7	2.61	1.827	0.6	3.21	1.926
	1	Sum WX	1.2345	1.5	Sum WX	4.3625	1.7	Sum WX	5.3645
D15									
alpha cut : 0	0.4	3	1.2	0.5	5.06	2.53	0.8	5.7	4.56
alpha cut : 0.5	0.6	2.55	1.53	0.8	4.4	3.52	0.8	5.7	4.56
alpha cut : 1	0.7	2.25	1.575	0.9	4.2	3.78	0.9	6	5.4
	1.7	Sum WX	4.305	2.2	Sum WX	9.83	2.5	Sum WX	14.52
D17									
alpha cut : 0	0.2	3.9	0.78	0.6	5	3	0.8	5.75	4.6
alpha cut : 0.5	0.3	3.75	1.125	0.7	4.85	3.395	0.8	5.75	4.6
alpha cut : 1	0.6	3.2	1.92	0.9	4.45	4.005	0.7	5.55	3.885
	1.1	Sum WX	3.825	2.2	Sum WX	10.4	2.3	Sum WX	13.085
D21									
alpha cut : 0	0.2	1.75	0.35	0.3	3.65	1.095	0.5	3.259	1.6295
alpha cut : 0.5	0.3	1.5	0.45	0.5	3.259	1.6295	0.6	3.45	2.07
alpha cut : 1	0.4	1.35	0.54	0.7	3.75	2.625	0.7	3.75	2.625
	0.9	Sum WX	1.34	1.5	Sum WX	5.3495	1.8	Sum WX	6.3245
D22									
alpha cut : 0	0.2	2.45	0.49	0.3	4.35	1.305	0.7	4.4	3.08
alpha cut : 0.5	0.2	2.45	0.49	0.5	3.936	1.968	0.7	4.4	3.08
alpha cut : 1	0.5	1.864	0.932	0.8	3.35	2.68	0.6	4.2	2.52
	0.9	Sum WX	1.912	1.6	Sum WX	5.953	2	Sum WX	8.68
D23									
alpha cut : 0	0.2	3.75	0.75	0.5	4.741	2.3705	0.8	5.18	4.144
alpha cut : 0.5	0.3	3.58	1.074	0.8	4.3	3.44	0.8	5.18	4.144
alpha cut : 1	0.5	3.299	1.6495	0.9	4.15	3.735	0.7	5	3.5
	1	Sum WX	3.4735	2.2	Sum WX	9.5455	2.3	Sum WX	11.788
	Sum WX = 48.849 Sum W = 21.7 Sum WX / Sum W = 2.25			**Sum WX = 130.005 Sum W = 34.6 Sum WX / Sum W = 3.76**			**Sum WX = 166.759 Sum W = 38.1 Sum WX / Sum W = 4.38**		

Impact on cost overrun :

Table 9.14 – 9.17 : Example of Possible alternatives of Fuzzy computation of the category of responses for the impact of risk factors on cost overrun.

Table 9.14 : Fuzzy computation for Feasibility stage of impact of risk factor on cost overrun

alpha cuts	Low			Medium			High		
	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX
F1									
alpha cut : 0	0.2	2.75	0.55	0.5	4.448	2.224	0.8	5.15	4.12
alpha cut : 0.5	0.3	2.5	0.75	0.6	4.22	2.532	0.8	5.15	4.12
alpha cut : 1	0.7	1.55	1.085	0.8	3.7	2.96	0.8	5.15	4.12
	1.2		**2.385**	**1.9**		**7.716**	**2.4**		**12.36**
F2									
alpha cut : 0	0.2	2.2	0.44	0.4	3.6	1.44	0.7	3.85	2.695
alpha cut : 0.5	0.3	2.05	0.615	0.6	3.25	1.95	0.7	3.85	2.695
alpha cut : 1	0.5	1.713	0.8565	0.7	3.05	2.135	0.6	3.6	2.16
	1		**1.9115**	**1.7**		**5.525**	**2**		**7.55**
F3									
alpha cut : 0	0.2	3.05	0.61	0.5	4.435	2.2175	0.7	4.8	3.36
alpha cut : 0.5	0.4	2.6	1.04	0.6	4.25	2.55	0.7	4.8	3.36
alpha cut : 1	0.7	2.05	1.435	0.9	3.6	3.24	0.8	5.05	4.04
	1.3		**3.085**	**2**		**8.0075**	**2.2**		**10.76**
F4									
alpha cut : 0	0.3	1.65	0.495	0.3	3.65	1.095	0.6	3.45	2.07
alpha cut : 0.5	0.3	1.65	0.495	0.4	3.45	1.38	0.6	3.45	2.07
alpha cut : 1	0.5	1.313	0.6565	0.8	2.65	2.12	0.6	3.45	2.07
	1.1		**1.6465**	**1.5**		**4.595**	**1.8**		**6.21**
F5									
alpha cut : 0	0.2	1.75	0.35	0.3	3	0.9	0.7	3.1	2.17
alpha cut : 0.5	0.2	1.75	0.35	0.4	2.9	1.16	0.6	2.9	1.74
alpha cut : 1	0.5	1.31	0.655	0.7	2.5	1.75	0.6	2.9	1.74
	0.9		**1.355**	**1.4**		**3.81**	**1.9**		**5.65**
	Sum WX = 10.383 **Sum W = 5.5** **Sum WX / Sum W = 1.89**			**Sum WX = 29.654** **Sum W = 8.5** **Sum WX / Sum W = 3.49**			**Sum WX = 42.53** **Sum W = 10.3** **Sum WX / Sum W = 4.13**		

Table 9.15 : Fuzzy computation for Project planning stage of impact of risk factor on cost overrun

alpha cuts	**Low**			**Medium**			**High**		
	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX
P1									
alpha cut : 0	0.3	3.98	1.194	0.6	4.78	2.868	0.8	5.2	4.16
alpha cut : 0.5	0.4	3.85	1.54	0.8	4.5	3.6	0.8	5.2	4.16
alpha cut : 1	0.6	3.6	2.16	0.9	4.4	3.96	0.7	5.1	3.57
	1.3		**4.894**	**2.3**		**10.43**	**2.3**		**11.89**
P2									
alpha cut : 0	0.3	3.2	0.96	0.6	4.3	2.58	0.8	4.95	3.96
alpha cut : 0.5	0.5	2.861	1.431	0.7	4.15	2.905	0.7	4.8	3.36
alpha cut : 1	0.7	2.5	1.75	0.8	4	3.2	0.7	4.8	3.36
	1.5		**4.141**	**2.1**		**8.685**	**2.2**		**10.68**
P3									
alpha cut : 0	0.2	2.2	0.44	0.5	3.414	1.707	0.7	3.75	2.625
alpha cut : 0.5	0.5	1.706	0.853	0.7	3.05	2.135	0.6	3.6	2.16
alpha cut : 1	0.6	1.5	0.9	0.7	3.05	2.135	0.6	3.6	2.16
	1.3		**2.193**	**1.9**		**5.977**	**1.9**		**6.945**
P4									
alpha cut : 0	0.3	2.48	0.744	0.5	4.167	2.084	0.7	4.65	3.255
alpha cut : 0.5	0.4	2.3	0.92	0.6	3.9	2.34	0.6	4.4	2.64
alpha cut : 1	0.6	1.88	1.128	0.7	3.7	2.59	0.7	4.65	3.255
	1.3		**2.792**	**1.8**		**7.014**	**2**		**9.15**
P5									
alpha cut : 0	0.2	3.45	0.69	0.6	4.45	2.67	0.8	5.1	4.08
alpha cut : 0.5	0.4	3.18	1.272	0.8	4.1	3.28	0.7	4.98	3.486
alpha cut : 1	0.6	2.8	1.68	0.8	4.1	3.28	0.8	5.1	4.08
	1.2		**3.642**	**2.2**		**9.23**	**2.3**		**11.646**
P6									
alpha cut : 0	0.4	3.5	1.4	0.6	4.95	2.97	0.8	5.67	4.536
alpha cut : 0.5	0.5	3.354	1.677	0.8	4.6	3.68	0.8	5.67	4.536
alpha cut : 1	0.7	3	2.1	0.9	4.4	3.96	0.9	5.8	5.22
	1.6		**5.177**	**2.3**		**10.61**	**2.5**		**14.292**
P7									
alpha cut : 0	0.3	3.85	1.155	0.6	4.95	2.97	0.9	5.75	5.175
alpha cut : 0.5	0.6	3.37	2.022	0.8	4.65	3.72	0.7	5.4	3.78
alpha cut : 1	0.6	3.37	2.022	0.8	4.65	3.72	0.8	5.57	4.456
	1.5		**5.199**	**2.2**		**10.41**	**2.4**		**13.411**

P8									
alpha cut : 0	0.2	1.9	0.38	0.5	3.268	1.634	0.7	3.65	2.555
alpha cut : 0.5	0.5	1.312	0.656	0.7	2.85	1.995	0.6	3.45	2.07
alpha cut : 1	0.5	1.312	0.656	0.8	2.65	2.12	0.6	3.45	2.07
	1.2		**1.692**	**2**		**5.749**	**1.9**		**6.695**
P9									
alpha cut : 0	0.2	3.16	0.632	0.5	4.414	2.207	0.8	4.96	3.968
alpha cut : 0.5	0.3	3	0.9	0.7	4.06	2.842	0.7	4.76	3.332
alpha cut : 1	0.5	2.646	1.323	0.8	3.86	3.088	0.6	4.6	2.76
	1		**2.855**	**2**		**8.137**	**2.1**		**10.06**
P10									
alpha cut : 0	0.2	1.63	0.326	0.5	2.968	1.484	0.7	3.4	2.38
alpha cut : 0.5	0.3	1.4	0.42	0.7	2.55	1.785	0.6	3.15	1.89
alpha cut : 1	0.4	1.2	0.48	0.7	2.55	1.785	0.5	2.968	1.484
	0.9		**1.226**	**1.9**		**5.054**	**1.8**		**5.754**
P11									
alpha cut : 0	0.2	2.58	0.516	0.4	4.05	1.62	0.6	4.38	2.628
alpha cut : 0.5	0.4	2.15	0.86	0.6	3.95	2.37	0.6	4.38	2.628
alpha cut : 1	0.5	1.912	0.956	0.7	3.7	2.59	0.6	4.38	2.628
	1.1		**2.332**	**1.7**		**6.58**	**1.8**		**7.884**
P12									
alpha cut : 0	0.4	3.37	1.348	0.6	4.97	2.982	0.8	5.8	4.64
alpha cut : 0.5	0.5	3.191	1.596	0.8	4.57	3.656	0.7	5.6	3.92
alpha cut : 1	0.7	2.77	1.939	0.7	4.77	3.339	0.6	5.4	3.24
	1.6		**4.883**	**2.1**		**9.977**	**2.1**		**11.8**
P13									
alpha cut : 0	0.3	1.65	0.495	0.5	2.883	1.442	0.7	3.25	2.275
alpha cut : 0.5	0.5	1.317	0.659	0.7	2.53	1.771	0.6	3.1	1.86
alpha cut : 1	0.6	1.13	0.678	0.8	2.4	1.92	0.6	3.1	1.86
	1.4		**1.832**	**2**		**5.133**	**1.9**		**5.995**
P14									
alpha cut : 0	0.4	2.04	0.816	0.6	4.44	2.664	0.7	5.44	3.808
alpha cut : 0.5	0.5	1.754	0.877	0.7	4.2	2.94	0.6	5.04	3.024
alpha cut : 1	0.6	1.44	0.864	0.8	3.84	3.072	0.5	4.786	2.393
	1.5		**2.557**	**2.1**		**8.676**	**1.8**		**9.225**
	Sum WX = 45.414 Sum W = 18.4 Sum WX / Sum W = 2.47			**Sum WX = 111.659 Sum W = 28.6 Sum WX / Sum W = 3.9**			**Sum WX = 135.427 Sum W = 29 Sum WX / Sum W = 4.67**		

Table 9.16 : Fuzzy computation for Requirement stage of impact of risk factor on cost overrun

alpha cuts	Low			Medium			High		
	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX
R1									
alpha cut : 0	0.3	3.46	1.038	0.5	4.857	2.4285	0.8	5.4	4.32
alpha cut : 0.5	0.5	3.123	1.5615	0.8	5.4	4.32	0.9	5.56	5.004
alpha cut : 1	0.7	2.76	1.932	0.9	5.56	5.004	0.9	5.56	5.004
	1.5		**4.5315**	**2.2**		**11.753**	**2.6**		**14.328**
R2									
alpha cut : 0	0.2	2.02	0.404	0.4	3.55	1.42	0.7	3.8	2.66
alpha cut : 0.5	0.3	1.82	0.546	0.7	3	2.1	0.8	2.8	2.24
alpha cut : 1	0.5	1.4	0.7	0.7	3	2.1	0.7	3.8	2.66
	1		**1.65**	**1.8**		**5.62**	**2.2**		**7.56**
R3									
alpha cut : 0	0.2	3.54	0.708	0.5	4.717	2.3585	0.8	5.2	4.16
alpha cut : 0.5	0.3	3.4	1.02	0.8	4.24	3.392	0.8	5.2	4.16
alpha cut : 1	0.6	2.9	1.74	0.8	4.24	3.392	0.9	5.4	4.86
	1.1		**3.468**	**2.1**		**9.1425**	**2.5**		**13.18**
R4									
alpha cut : 0	0.2	1.7	0.34	0.4	2.8	1.12	0.7	2.95	2.065
alpha cut : 0.5	0.3	2.9	0.87	0.7	2.4	1.68	0.7	2.95	2.065
alpha cut : 1	0.4	2.8	1.12	0.6	2.53	1.518	0.6	2.8	1.68
	0.9		**2.33**	**1.7**		**4.318**	**2**		**5.81**
R5									
alpha cut : 0	0.2	1.51	0.302	0.4	2.91	1.164	0.6	3	1.8
alpha cut : 0.5	0.2	1.51	0.302	0.7	2.5	1.75	0.8	3.31	2.648
alpha cut : 1	0.6	0.85	0.51	0.7	2.5	1.75	0.7	2.45	1.715
	1		**1.114**	**1.8**		**4.664**	**2.1**		**6.163**
R6									
alpha cut : 0	0.3	2.93	0.879	0.4	5.53	2.212	0.8	4.43	3.544
alpha cut : 0.5	0.4	2.65	1.06	0.7	4.73	3.311	0.8	4.43	3.544
alpha cut : 1	0.6	2.1	1.26	0.9	4.13	3.717	0.9	6.43	5.787
	1.3		**3.199**	**2**		**9.24**	**2.5**		**12.875**
R7									
alpha cut : 0	0.2	2.8	0.56	0.4	4.38	1.752	0.6	3.98	2.388
alpha cut : 0.5	0.4	2.45	0.98	0.5	4.221	2.1105	0.7	4.6	3.22
alpha cut : 1	0.6	2.05	1.23	0.8	3.6	2.88	0.7	4.6	3.22
	1.2		**2.77**	**1.7**		**6.7425**	**2**		**8.828**
	Sum WX = 19.063 Sum W = 8 Sum WX / Sum W = 2.38			**Sum WX = 51.479 Sum W = 13.3 Sum WX / Sum W = 3.87**			**Sum WX = 68.744 Sum W = 15.9 Sum WX / Sum W = 4.32**		

Table 9.17 : Fuzzy computation for Development stage of impact of risk factor on cost overrun

alpha cuts	Low			Medium			High		
	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX
D1									
alpha cut : 0	0.2	2.6	0.52	0.4	4.76	1.904	0.7	5.1	3.57
alpha cut : 0.5	0.4	2.06	0.824	0.6	4.26	2.556	0.7	5.1	3.57
alpha cut : 1	0.6	1.56	0.936	0.7	3.96	2.772	0.5	4.556	2.278
	1.2		**2.28**	**1.7**		**7.232**	**1.9**		**9.418**
D2									
alpha cut : 0	0.3	2.95	0.885	0.5	4.722	2.361	0.8	5.42	4.336
alpha cut : 0.5	0.5	2.518	1.259	0.8	4.05	3.24	0.7	5.15	3.605
alpha cut : 1	0.7	2.1	1.47	0.9	3.82	3.438	0.6	4.95	2.97
	1.5		**3.614**	**2.2**		**9.039**	**2.1**		**10.911**
D3									
alpha cut : 0	0.3	1.7	0.51	0.3	3.65	1.095	0.6	3.5	2.1
alpha cut : 0.5	0.4	1.5	0.6	0.7	2.85	1.995	0.7	3.72	2.604
alpha cut : 1	0.6	1.124	0.6744	0.7	2.85	1.995	0.6	3.5	2.1
	1.3		**1.7844**	**1.7**		**5.085**	**1.9**		**6.804**
D4									
alpha cut : 0	0.3	1.75	0.525	0.4	3.87	1.548	0.7	4.15	2.905
alpha cut : 0.5	0.5	1.407	0.7035	0.6	3.4	2.04	0.6	3.87	2.322
alpha cut : 1	0.5	1.407	0.7035	0.7	3.15	2.205	0.6	3.87	2.322
	1.3		**1.932**	**1.7**		**5.793**	**1.9**		**7.549**
D5									
alpha cut : 0	0.2	2.85	0.57	0.4	4.6	1.84	0.8	5.05	4.04
alpha cut : 0.5	0.4	2.4	0.96	0.7	3.95	2.765	0.7	4.85	3.395
alpha cut : 1	0.7	1.7	1.19	0.7	3.95	2.765	0.5	4.393	2.1965
	1.3		**2.72**	**1.8**		**7.37**	**2**		**9.6315**
D6									
alpha cut : 0	0.3	2.05	0.615	0.4	3.85	1.54	0.8	4.2	3.36
alpha cut : 0.5	0.5	1.697	0.8485	0.8	3.05	2.44	0.6	3.85	2.31
alpha cut : 1	0.5	1.697	0.8485	0.7	3.2	2.24	0.5	3.643	1.8215
	1.3		**2.312**	**1.9**		**6.22**	**1.9**		**7.4915**
D7									
alpha cut : 0	0.3	1.65	0.495	0.4	3.4	1.36	0.7	3.65	2.555
alpha cut : 0.5	0.5	1.342	0.671	0.7	2.85	1.995	0.6	3.45	2.07
alpha cut : 1	0.6	1.15	0.69	0.7	2.85	1.995	0.5	3.278	1.639
	1.4		**1.856**	**1.8**		**5.35**	**1.8**		**6.264**
D8									
alpha cut : 0	0.2	1.7	0.34	0.3	3.21	0.963	0.8	3.4	2.72
alpha cut : 0.5	0.4	1.41	0.564	0.6	2.71	1.626	0.6	3.05	1.83
alpha cut : 1	0.6	1.1	0.66	0.7	2.51	1.757	0.5	2.885	1.4425
	1.2		**1.564**	**1.6**		**4.346**	**1.9**		**5.9925**
D9									
alpha cut : 0	0.2	1.72	0.344	0.3	3.52	1.056	0.6	3.4	2.04
alpha cut : 0.5	0.4	1.32	0.528	0.7	2.72	1.904	0.6	3.4	2.04
alpha cut : 1	0.5	1.14	0.57	0.7	2.72	1.904	0.6	3.4	2.04
	1.1		**1.442**	**1.7**		**4.864**	**1.8**		**6.12**

D10									
alpha cut : 0	0.3	1.65	0.495	0.3	3.65	1.095	0.7	3.72	2.604
alpha cut : 0.5	0.5	1.307	0.6535	0.7	2.85	1.995	0.6	3.5	2.1
alpha cut : 1	0.5	1.307	0.6535	0.6	3.1	1.86	0.5	3.273	1.6365
	1.3		**1.802**	**1.6**		**4.95**	**1.8**		**6.3405**
D11									
alpha cut : 0	0.3	3.09	0.927	0.4	5.69	2.276	0.8	6.29	5.032
alpha cut : 0.5	0.6	2.25	1.35	0.8	4.55	3.64	0.8	6.29	5.032
alpha cut : 1	0.6	2.25	1.35	0.7	4.89	3.423	0.6	5.69	3.414
	1.5		**3.627**	**1.9**		**9.339**	**2.2**		**13.478**
D12									
alpha cut : 0	0.3	1.9	0.57	0.4	3.45	1.38	0.7	3.65	2.555
alpha cut : 0.5	0.4	1.72	0.688	0.7	2.92	2.044	0.7	3.65	2.555
alpha cut : 1	0.6	1.4	0.84	0.8	2.72	2.176	0.6	3.45	2.07
	1.3		**2.098**	**1.9**		**5.6**	**2**		**7.18**
D13									
alpha cut : 0	0.3	2.5	0.75	0.4	4.4	1.76	0.8	4.83	3.864
alpha cut : 0.5	0.4	2.3	0.92	0.6	4	2.4	0.7	4.63	3.241
alpha cut : 1	0.5	2.085	1.0425	0.8	3.55	2.84	0.4	4	1.6
	1.2		**2.7125**	**1.8**		**7**	**1.9**		**8.705**
D14									
alpha cut : 0	0.3	1.5	0.45	0.4	3.4	1.36	0.6	3.4	2.04
alpha cut : 0.5	0.5	1.072	0.536	0.7	2.75	1.925	0.6	3.4	2.04
alpha cut : 1	0.5	1.072	0.536	0.6	3	1.8	0.6	3.4	2.04
	1.3		**1.522**	**1.7**		**5.085**	**1.8**		**6.12**
D15									
alpha cut : 0	0.3	3.14	0.942	0.4	5.3	2.12	0.8	5.75	4.6
alpha cut : 0.5	0.5	2.718	1.359	0.8	4.34	3.472	0.8	5.75	4.6
alpha cut : 1	0.6	2.5	1.5	0.9	4.1	3.69	0.6	5.3	3.18
	1.4		**3.801**	**2.1**		**9.282**	**2.2**		**12.38**
D17									
alpha cut : 0	0.3	2.21	0.663	0.3	5.2	1.56	0.7	5.2	3.64
alpha cut : 0.5	0.4	2	0.8	0.5	4.578	2.289	0.6	4.85	2.91
alpha cut : 1	0.7	1.1	0.77	0.8	3.75	3	0.4	4.25	1.7
	1.4		**2.233**	**1.6**		**6.849**	**1.7**		**8.25**
D19									
alpha cut : 0	0.3	1.44	0.432	0.3	2.95	0.885	0.7	3.048	2.1336
alpha cut : 0.5	0.4	1.35	0.54	0.5	2.701	1.3505	0.6	2.848	1.7088
alpha cut : 1	0.6	1.048	0.6288	0.8	2.24	1.792	0.4	2.55	1.02
	1.3		**1.6008**	**1.6**		**4.0275**	**1.7**		**4.8624**
D22									
alpha cut : 0	0.2	2.12	0.424	0.5	3.666	1.833	0.6	3.92	2.352
alpha cut : 0.5	0.4	1.65	0.66	0.4	3.85	1.54	0.6	3.92	2.352
alpha cut : 1	0.5	1.434	0.717	0.6	3.45	2.07	0.4	3.5	1.4
	1.1		**1.801**	**1.5**		**5.443**	**1.6**		**6.104**
D23									
alpha cut : 0	0.4	3.3	1.32	0.5	4.706	2.353	0.9	5.4	4.86
alpha cut : 0.5	0.4	3.3	1.32	0.8	4.208	3.3664	0.7	5.1	3.57
alpha cut : 1	0.7	2.81	1.967	0.9	4.1	3.69	0.4	4.6	1.84
	1.5		**4.607**	**2.2**		**9.4094**	**2**		**10.27**
	Sum WX = 45.309			**Sum WX = 122.284**			**Sum WX = 153.872**		
	Sum W = 24.9			**Sum W = 33.7**			**Sum W = 36.1**		
	Sum WX / Sum W = 1.82			**Sum WX / Sum W = 3.63**			**Sum WX / Sum W = 4.26**		

The combination of different possibilities and scenarios of the MBF are calculated for each risk factor using the different level of significance of linguistic variables. The Low, Moderate and High level of risk factors for each stage can be calculated with the summation of the alternatives. The results of the computation is presented as the scores of stages on a spider net in Figure 9.15 and Figure 9.16.

The W*F(x) for each risk factor is calculated, and then, the summation of W*F(x) is divided to the summation of the weights (W).

Table 9.18 : Summation of fuzzy computation for each stage.

Stage	Low		Moderate		High	
	Likelihood occurrence	Impact of risk factor on cost overrun	Likelihood occurrence	Impact of risk factor on cost overrun	Likelihood occurrence	Impact of risk factor on cost overrun
Feasibility study	2.37	1.89	3.86	3.49	4.28	4.13
Project Planning	2.58	2.47	3.9	3.9	4.66	4.67
Requirement	2.33	2.38	3.96	3.87	4.43	4.32
Development	2.25	1.82	3.76	3.63	4.37	4.26

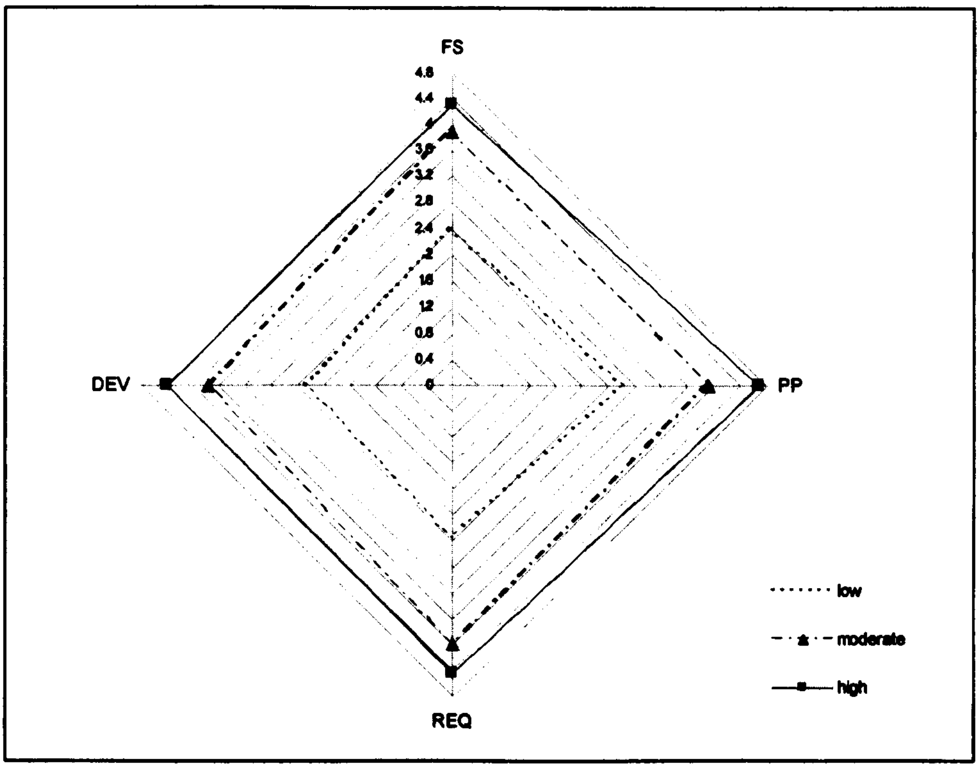


Figure 9.15 : Low, Moderate and High category level of Likelihood occurrence of software project risk for each stage. (values obtained from Table 9.17)

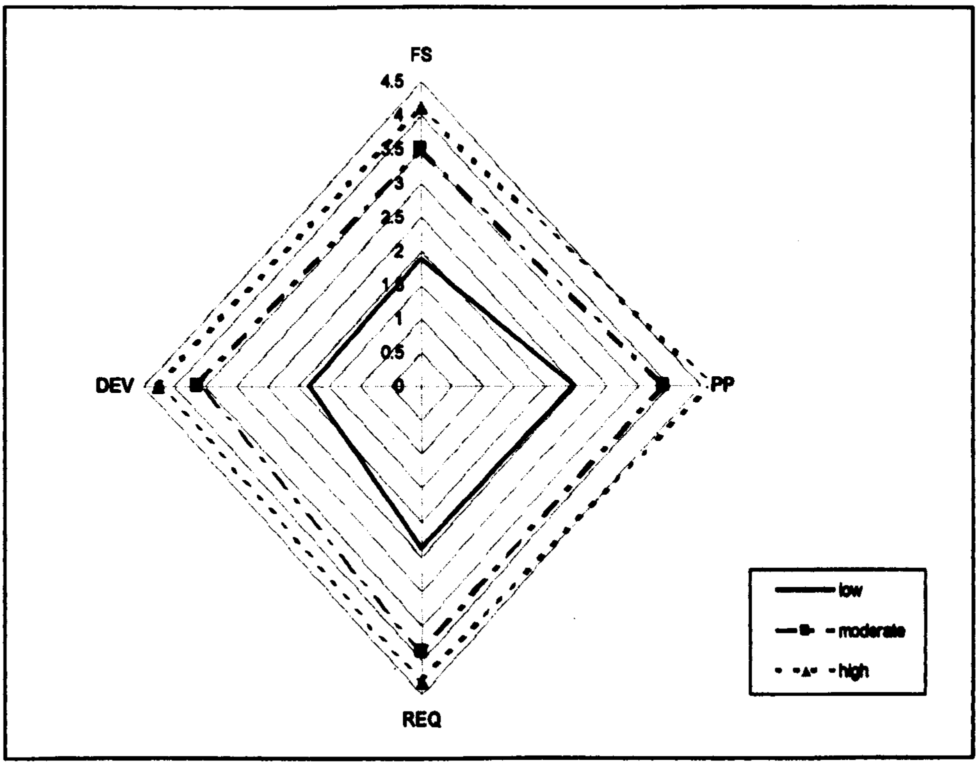


Figure 9.16 : Low, Moderate and High category level of impact of risk factor on cost overrun for each stage. (values obtained from Table 9.17)

From the calculation of the fuzzy computation of Low, Moderate and High category levels for risk factors in each stage, the results were illustrated in diagrams shown in Figure 9.15 and Figure 9.16. In the diagram, the Low, Moderate and High category for Likelihood of occurrence and risk impact on cost overrun were shown on an axis which used radius of circles as the scale of measurement. The points for each stage are connected to each other that formed a looped line around the diagram. These connected lines represent the Likelihood occurrence and risk impact on cost overrun, for the Low, Moderate and High category levels for risk factors of the stages.

Speculatively, the Low category levels of risks factors can also be represented as the Acceptable level of risks, the moderate category as the High risk level and the high category as the Very high level of risks. In other words, the Acceptable level of risks would be a risk profile that is acceptable for the software project to be considered feasible to proceed, when taking into account the related risk factors involved in the stages.

For the High risk level, the risk profile would probably mean that the software project can still go ahead despite the high risk profile but with extra precautionary measures. A more detail analysis of the risk factors might also be advisable before any decision being made. Eventhough, the project did proceed, the risk mitigation strategies should also be high on the agenda.

As for the Very high risk level, the risk profile might suggest that it might not be worth taking on the project as the level of risk is too high in terms of the occurrence of risk and its impact on cost overrun.

9.10. Fuzzy Model application

In practice, the model can be used from the responses of a real life project or respondents. The calculation and computation of the risk factors from the real life project responses were applied to the model. The risk profile for the project can be plotted, and the risk profile of particular stages of the life cycle can be evaluated. This will assist in the decision making processes of handling the risk factors and identifying the appropriate risk mitigation strategies.

However, in order to demonstrate the application of this modeling technique, a few responses from the survey can be used as a hypothetical example, rather than real life project. For example, consider one respondent as 'one project', and a few respondents as a 'few projects'. For example, consider 4 projects as Project 1, Project 2, Project 3 and Project 4. Consider Project 1 and Project 2 for the score of the likelihood occurrence and Project 3 and Project 4 for the score of risk impact on cost overrun.

Example 1 : Project 1 - Likelihood occurrence

Scoring for Likelihood occurrence of risk factors;
1-don't know; 2-unlikely; 3-likely; 4-highly likely; 5-very highly likely

Table 9.19 : Example of the fuzzy model application scores for Project 1 (Likelihood occurrence)

		The likelihood occurrence of the risk factors in the software project	Score	Degree of belief	Degree of significance Fij(x)	∑WjFij(x)/∑Wj
Feasibility Study stage	F1	Inproper justification of cost benefit analysis and evaluation criteria from feasibility study	5	0.5	0.28	2.01
	F2	Too narrow focus on the technical IT issues	3	0.4	0.85	
	F3	Overlooked the management and business impact issues	4	0.5	0.75	
	F4	Wrong justification of investment alternatives and opportunity cost	2	0.5	0.8	
	F5	Inappropriate technology chosen from the feasibility study	2	0.5	0.8	
Project planning stage	P1	Unclear project scope + objectives	5	0.8	0.63	2.20
	P2	Undefined project success criteria	3	0.7	0.4	
	P3	Lack of quality control procedure and mechanism	1	0.5	0.2	
	P4	Project milestones for stages not well established	2	0.6	0.2	
	P5	Improper change management planning	4	0.6	0.77	
	P6	Inaccurate estimate of resources	5	0.7	0.68	
	P7	Unrealistic project schedule	5	0.7	0.63	
	P8	Inadequate detail work breakdown structure	3	0.6	0.8	
	P9	Critical and non-critical activities of project not identified	3	0.7	0.65	
	P10	Project management & development team not properly set up	1	0.6	0.55	
	P11	Unclear line of decision making authority throughout the project	4	0.7	1	
	P12	Lack of contingency plan/back up	4	0.7	0.72	
	P13	System conversion method not well planned	2	0.6	0.3	
	P14	Improper planning of timeframe for project reviews and updating	3	0.5	0.9	
Requirement stage	R1	Unclear and inadequate identification of systems requirements	5	0.7	0.4	2.27
	R2	Incorrect systems requirements	3	0.4	0.7	
	R3	Misinterpretations of the systems requirements	4	0.7	0.6	
	R4	Conflicting system requirements	2	0.4	0.8	
	R5	Gold plating or unnecessary functions and requirements	1	0.5	0.5	
	R6	Inadequate validation of the requirements	4	0.6	0.9	
	R7	Lack of users involvement in requirement stage	4	0.5	0.83	
Development stage	D1	Inproper handover from the requirement team	2	0.4	0.45	0.97
	D2	Inappropriate development methodology used	3	0.8	0.33	
	D3	Unsuitable working model and prototype	2	0.4	0.15	
	D4	Programming language and CASE tool selected not adequate	2	0.8	0.2	
	D5	High level of technical complexities	2	0.8	0.1	
	D6	Project involves the use of new technology	3	0.8	0.68	
	D7	Difficulty in defining the input and output of system	1	0.5	0.1	
	D8	Immature technology	2	0.5	0.9	
	D9	Technological advancements and changes	3	0.4	0.4	
	D10	Failures and inconsistencies of unit/modules test results	1	0.5	0.4	
	D11	Failure of user acceptance test	2	0.6	0.35	
	D12	Time consuming for testing	2	0.8	0.05	
	D13	Resources shifted from project due to organisational priorities	3	0.5	0.85	
	D14	Changes in management of organisation during development	3	0.5	0.5	
	D15	Lack of users involvement and commitment	3	0.8	0.65	
	D17	Ineffective communication within development team members	3	0.7	0.28	
	D21	Inexperienced team members	1	0.5	0.45	
	D22	Lack of commitment to project among development team members	2	0.5	0.6	
	D23	Ineffective and inexperienced project manager	3	0.8	0.33	

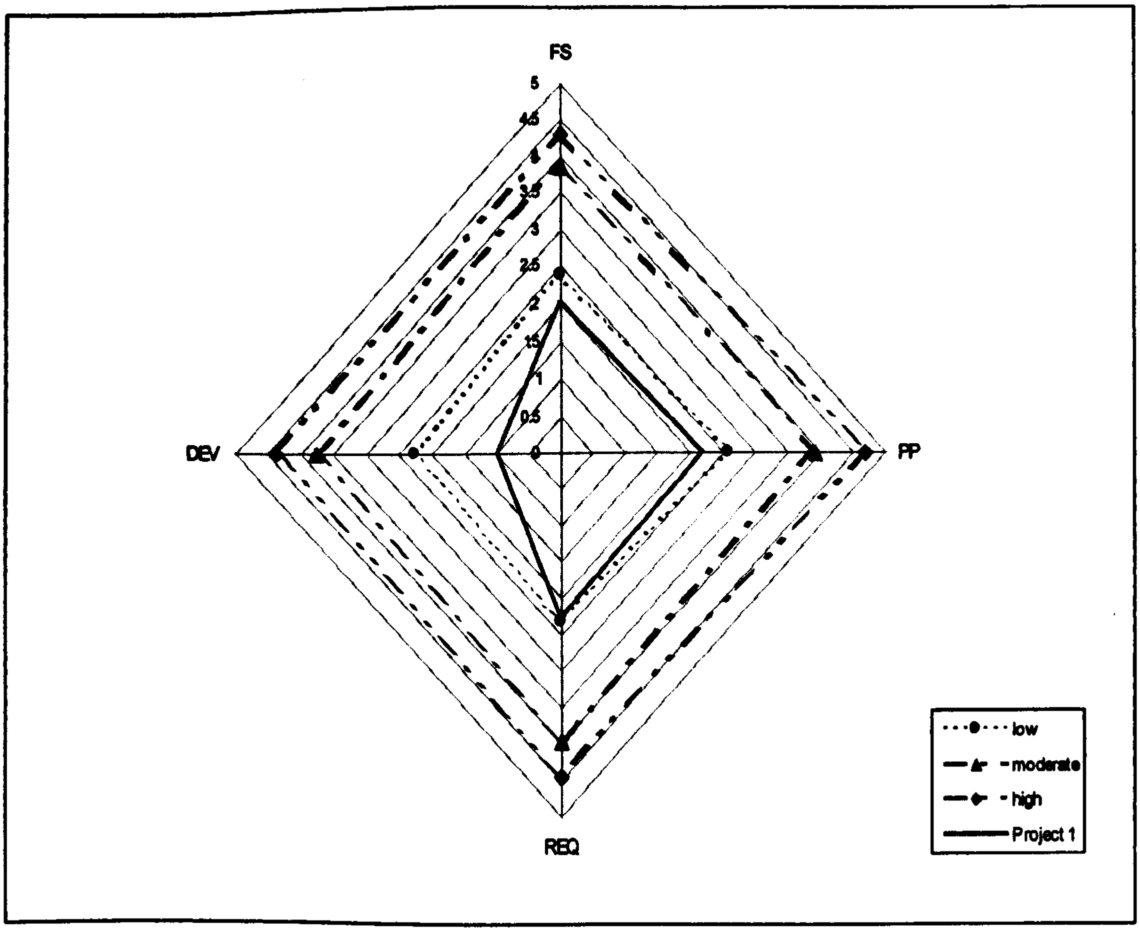


Figure 9.17 : The fuzzy model application for Project 1 (Likelihood occurrence)

The dotted line in Figure 9.17 shows the risk level profile from the model. The risk profile for 'Project 1' was also shown. As a simple guideline, when the risk profile for 'Project 1' do not exceed the Low (Acceptable) risk profile of the model, a more straightforward decision to proceed with the software project can be made. But if the risk profile of 'Project 1', exceed the model Acceptable profile, careful considerations were needed before reaching to any decision, as to whether to proceed with the project or not.

For example, from the profiles of 'Project 1' in Figure 9.17, it can be seen that the level of risks for the feasibility study and project planning stage were lower than the 'low category' risk profile of the model. The level of risk during the development stage for 'Project 1' is far much lower than the 'low category' of the model. But, the level of risks for the requirement stage is almost the same level of the 'low category'. Based on this profile, it can be said that for 'Project 1', it has much very much lower risk during the development stage.

Example 2 : Project 2 - Likelihood occurrence

Scoring for Likelihood occurrence of risk factors;
1-don't know; 2-unlikely; 3-likely; 4-highly likely; 5-very highly likely

Table 9.20 : Example of the fuzzy model application scores for Project 2 (Likelihood occurrence)

		The likelihood occurrence of the risk factors in the software project	Score	Degree of belief	Degree of significance Fij(x)	ΣWjFij(x)/ΣWj
Feasibility Study stage	F1	Inproper justification of cost benefit analysis and evaluation criteria from feasibility study	2	0.5	0.15	
	F2	Too narrow focus on the technical IT issues	3	0.4	0.8	0.96
	F3	Overlooked the management and business impact issues	2	0.5	0.3	
	F4	Wrong justification of investment alternatives and opportunity cost	2	0.5	0.8	
	F5	Inappropriate technology chosen from the feasibility study	1	0.5	0.2	
Project planning stage	P1	Unclear project scope + objectives	3	0.8	0.45	
	P2	Undefined project success criteria	3	0.7	0.58	1.34
	P3	Lack of quality control procedure and mechanism	3	0.5	0.8	
	P4	Project milestones for stages not well established	2	0.6	0.8	
	P5	Improper change management planning	3	0.6	0.7	
	P6	Inaccurate estimate of resources	3	0.7	0.15	
	P7	Unrealistic project schedule	3	0.7	0.1	
	P8	Inadequate detail work breakdown structure	1	0.6	0.2	
	P9	Critical and non-critical activities of project not identified	5	0.7	0.38	
	P10	Project management & development team not properly set up	3	0.6	0.4	
	P11	Unclear line of decision making authority throughout the project	3	0.7	0.45	
	P12	Lack of contingency plan/back up	4	0.7	0.7	
	P13	System conversion method not well planned	2	0.6	0.3	
	P14	Improper planning of timeframe for project reviews and updating	2	0.5	0.35	
Requirement stage	R1	Unclear and inadequate identification of systems requirements	3	0.7	0.3	
	R2	Incorrect systems requirements	3	0.4	0.7	1.13
	R3	Misinterpretations of the systems requirements	3	0.7	0.8	
	R4	Conflicting system requirements	2	0.4	0.8	
	R5	Gold plating or unnecessary functions and requirements	1	0.5	0.5	
	R6	Inadequate validation of the requirements	2	0.6	0.1	
	R7	Lack of users involvement in requirement stage	1	0.5	0.28	
Development stage	D1	Inproper handover from the requirement team	2	0.4	0.45	
	D2	Inappropriate development methodology used	3	0.8	0.33	
	D3	Unsuitable working model and prototype	3	0.4	0.7	
	D4	Programming language and CASE tool selected not adequate	2	0.8	0.75	1.08
	D5	High level of technical complexities	5	0.8	0.4	
	D6	Project involves the use of new technology	1	0.8	0.35	
	D7	Difficulty in defining the input and output of system	3	0.5	0.9	
	D8	Immature technology	2	0.5	0.9	
	D9	Technological advancements and changes	1	0.4	0.4	
	D10	Failures and inconsistencies of unit/modules test results	3	0.5	0.6	
	D11	Failure of user acceptance test	2	0.6	0.35	
	D12	Time consuming for testing	2	0.8	0.05	
	D13	Resources shifted from project due to organisational priorities	5	0.5	0.15	
	D14	Changes in management of organisation during development	1	0.5	0.5	
	D15	Lack of users involvement and commitment	2	0.8	0.1	
	D17	Ineffective communication within development team members	3	0.7	0.3	
	D21	Inexperienced team members	3	0.5	0.6	
	D22	Lack of commitment to project among development team members	2	0.5	0.6	
	D23	Ineffective and inexperienced project manager	3	0.8	0.3	

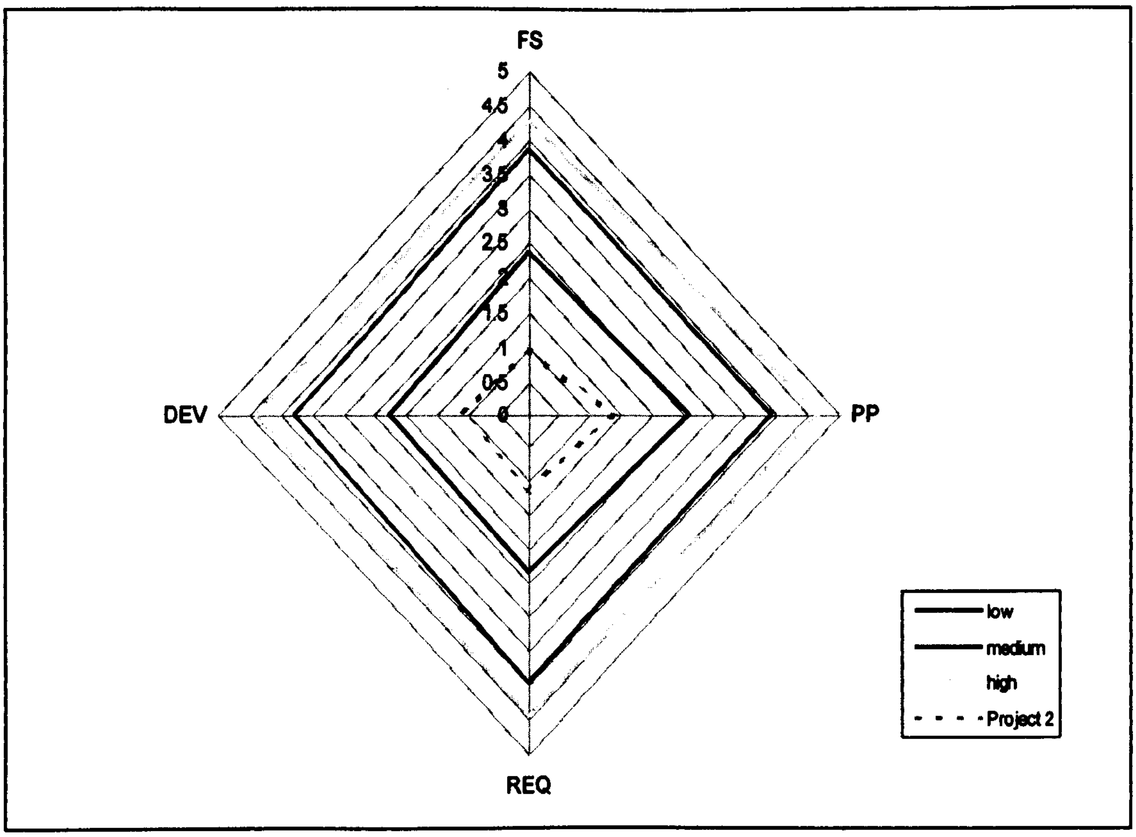


Figure 9.18 : The fuzzy model application for Project 2 (Likelihood occurrence)

From the profiles of 'Project 2' in Figure 9.18, it can be seen that the level of risks for all the stages were lower than the 'low category' risk profile of the model. This shows that Project 2 has a very low risk profile, which may also mean the higher chance of success.

Example 3 : Project 3 – Impact of risk factors on cost overrun

Scoring for impact of risk factors on cost overrun
1-very low(1-10% overrun); 2-low (11-20% overrun); 3-moderate (21-30% overrun); 4-high (31-40% overrun); 5-very high (>40% overrun)

Table 9.21 : Example of the fuzzy model application scores of Project 3 (Impact of risk factors on cost overrun)

		The likelihood occurrence of the risk factors in the software project	Score	Degree of belief	Degree of significance Fij(x)	∑WjFij(x)/∑Wj
Feasibility Study stage	F1	Inproper justification of cost benefit analysis and evaluation criteria from feasibility study	3	0.8	0.9	2.05
	F2	Too narrow focus on the technical IT issues	2	0.7	0.65	
	F3	Overlooked the management and business impact issues	3	0.9	0.8	
	F4	Wrong justification of investment alternatives and opportunity cost	3	0.8	0.6	
	F5	Inappropriate technology chosen from the feasibility study	2	0.7	0.95	
Project planning stage	P1	Unclear project scope + objectives	5	0.9	0.38	2.11
	P2	Undefined project success criteria	5	0.8	0.18	
	P3	Lack of quality control procedure and mechanism	2	0.7	0.65	
	P4	Project milestones for stages not well established	3	0.7	0.9	
	P5	Improper change management planning	4	0.8	0.85	
	P6	Inaccurate estimate of resources	5	0.9	0.55	
	P7	Unrealistic project schedule	4	0.8	0.8	
	P8	Inadequate detail work breakdown structure	4	0.8	0.1	
	P9	Critical and non-critical activities of project not identified	5	0.8	0.15	
	P10	Project management & development team not properly set up	2	0.7	0.95	
	P11	Unclear line of decision making authority throughout the project	3	0.7	0.98	
	P12	Lack of contingency plan/back up	4	0.7	0.9	
	P13	System conversion method not well planned	3	0.8	0.4	
	P14	Improper planning of timeframe for project reviews and updating	4	0.8	0.7	
Requirement stage	R1	Unclear and inadequate identification of systems requirements	4	0.9	0.98	2.06
	R2	Incorrect systems requirements	4	0.7	0.2	
	R3	Misinterpretations of the systems requirements	3	0.8	0.45	
	R4	Conflicting system requirements	3	0.6	0.3	
	R5	Gold plating or unnecessary functions and requirements	2	0.7	0.95	
	R6	Inadequate validation of the requirements	4	0.9	0.95	
	R7	Lack of users involvement in requirement stage	2	0.8	0.4	
Development stage	D1	Inproper handover from the requirement team	3	0.7	0.93	2.12
	D2	Inappropriate development methodology used	5	0.9	0.35	
	D3	Unsuitable working model and prototype	2	0.7	0.9	
	D4	Programming language and CASE tool selected not adequate	3	0.7	0.8	
	D5	High level of technical complexities	5	0.7	0.23	
	D6	Project involves the use of new technology	4	0.7	0.3	
	D7	Difficulty in defining the input and output of system	2	0.7	0.85	
	D8	Immature technology	3	0.7	0.4	
	D9	Technological advancements and changes	4	0.7	0.2	
	D10	Failures and inconsistencies of unit/modules test results	2	0.6	0.9	
	D11	Failure of user acceptance test	4	0.7	1	
	D12	Time consuming for testing	3	0.8	0.65	
	D13	Resources shifted from project due to organisational priorities	3	0.8	0.92	
	D14	Changes in management of organisation during development	4	0.6	0.1	
	D15	Lack of users involvement and commitment	5	0.9	0.55	
	D17	Ineffective communication within development team members	3	0.8	0.95	
	D19	Inadequately trained development team members	2	0.8	0.95	
	D22	Lack of commitment to project among development team members	3	0.6	0.8	
	D23	Ineffective and inexperienced project manager	4	0.9	0.95	

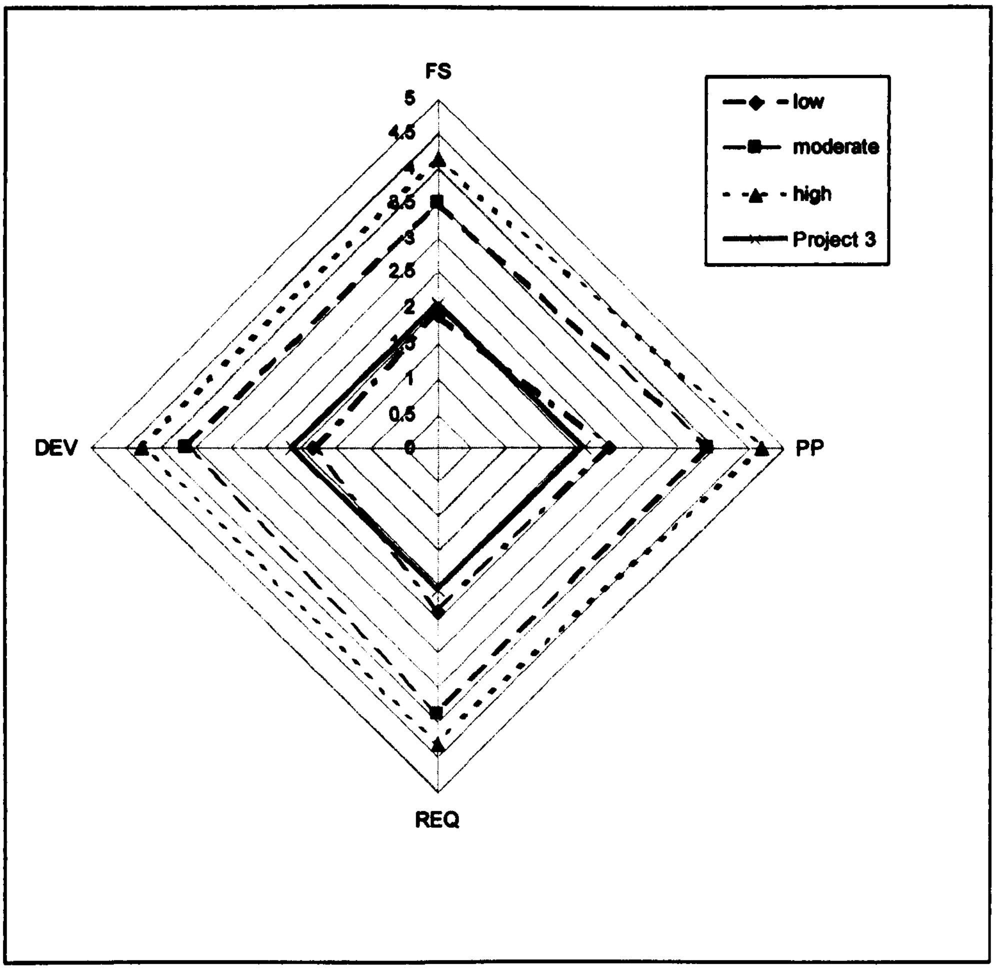


Figure 9.19 : The fuzzy model application for Project 3 (Impact of risk factor on cost overrun)

From the profiles of 'Project 3' in Figure 9.19, it can be seen that the level of risks for the feasibility study stage of 'Project 3' is near the same level as the 'low category' risk profile of the model, but the levels of risk during the project planning stage and requirement stage of 'Project 3' is lower than the 'low category' of the model. The level of risks for the development stage of 'Project 3' is higher than the 'moderate category'.

Example 4 : Project 4 – Impact of risk factors on cost overrun

Scoring for impact of risk factors on cost overrun;
1-very low(1-10% overrun); 2-low (11-20% overrun); 3-moderate (21-30% overrun);
4-high (31-40% overrun); 5-very high (>40% overrun)

Table 9.22 : Example of the fuzzy model application scores of Project 4 (Impact of risk factors on cost overrun)

		The likelihood occurrence of the risk factors in the software project	Score	Degree of belief	Degree of significance Fij(x)	∑WjFij(x)/∑Wj
Feasibility Study stage	F1	Inproper justification of cost benefit analysis and evaluation criteria from feasibility study	2	0.8	0.48	1.18
	F2	Too narrow focus on the technical IT issues	1	0.7	0.2	
	F3	Overlooked the management and business impact issues	3	0.9	0.8	
	F4	Wrong justification of investment alternatives and opportunity cost	1	0.8	0.35	
	F5	Inappropriate technology chosen from the feasibility study	2	0.7	0.9	
Project planning stage	P1	Unclear project scope + objectives	4	0.9	0.7	2.8
	P2	Undefined project success criteria	4	0.8	0.8	
	P3	Lack of quality control procedure and mechanism	3	0.7	0.7	
	P4	Project milestones for stages not well established	3	0.7	0.9	
	P5	Improper change management planning	4	0.8	0.85	
	P6	Inaccurate estimate of resources	4	0.9	0.9	
	P7	Unrealistic project schedule	4	0.8	0.8	
	P8	Inadequate detail work breakdown structure	2	0.8	0.95	
	P9	Critical and non-critical activities of project not identified	4	0.8	0.73	
	P10	Project management & development team not properly set up	2	0.7	0.95	
	P11	Unclear line of decision making authority throughout the project	3	0.7	0.98	
	P12	Lack of contingency plan/back up	4	0.7	0.9	
	P13	System conversion method not well planned	2	0.8	0.95	
	P14	Improper planning of timeframe for project reviews and updating	4	0.8	0.7	
Requirement stage	R1	Unclear and inadequate identification of systems requirements	4	0.9	0.98	2.9
	R2	Incorrect systems requirements	2	0.7	0.8	
	R3	Misinterpretations of the systems requirements	4	0.8	0.95	
	R4	Conflicting system requirements	2	0.6	0.9	
	R5	Gold plating or unnecessary functions and requirements	2	0.7	0.95	
	R6	Inadequate validation of the requirements	4	0.9	0.95	
	R7	Lack of users involvement in requirement stage	3	0.8	0.9	
Development stage	D1	Inproper handover from the requirement team	3	0.7	0.93	2.44
	D2	Inappropriate development methodology used	4	0.9	0.85	
	D3	Unsuitable working model and prototype	2	0.7	0.9	
	D4	Programming language and CASE tool selected not adequate	3	0.7	0.8	
	D5	High level of technical complexities	3	0.7	0.9	
	D6	Project involves the use of new technology	3	0.7	0.9	
	D7	Difficulty in defining the input and output of system	2	0.7	0.85	
	D8	Immature technology	3	0.7	0.4	
	D9	Technological advancements and changes	4	0.7	0.2	
	D10	Failures and inconsistencies of unit/modules test results	2	0.6	0.9	
	D11	Failure of user acceptance test	4	0.7	1	
	D12	Time consuming for testing	3	0.8	0.65	
	D13	Resources shifted from project due to organisational priorities	3	0.8	0.92	
	D14	Changes in management of organisation during development	3	0.6	0.6	
	D15	Lack of users involvement and commitment	5	0.9	0.55	
	D17	Ineffective communication within development team members	3	0.8	0.95	
	D19	Inadequately trained development team members	2	0.8	0.95	
	D22	Lack of commitment to project among development team members	3	0.6	0.8	
	D23	Ineffective and inexperienced project manager	4	0.9	0.95	

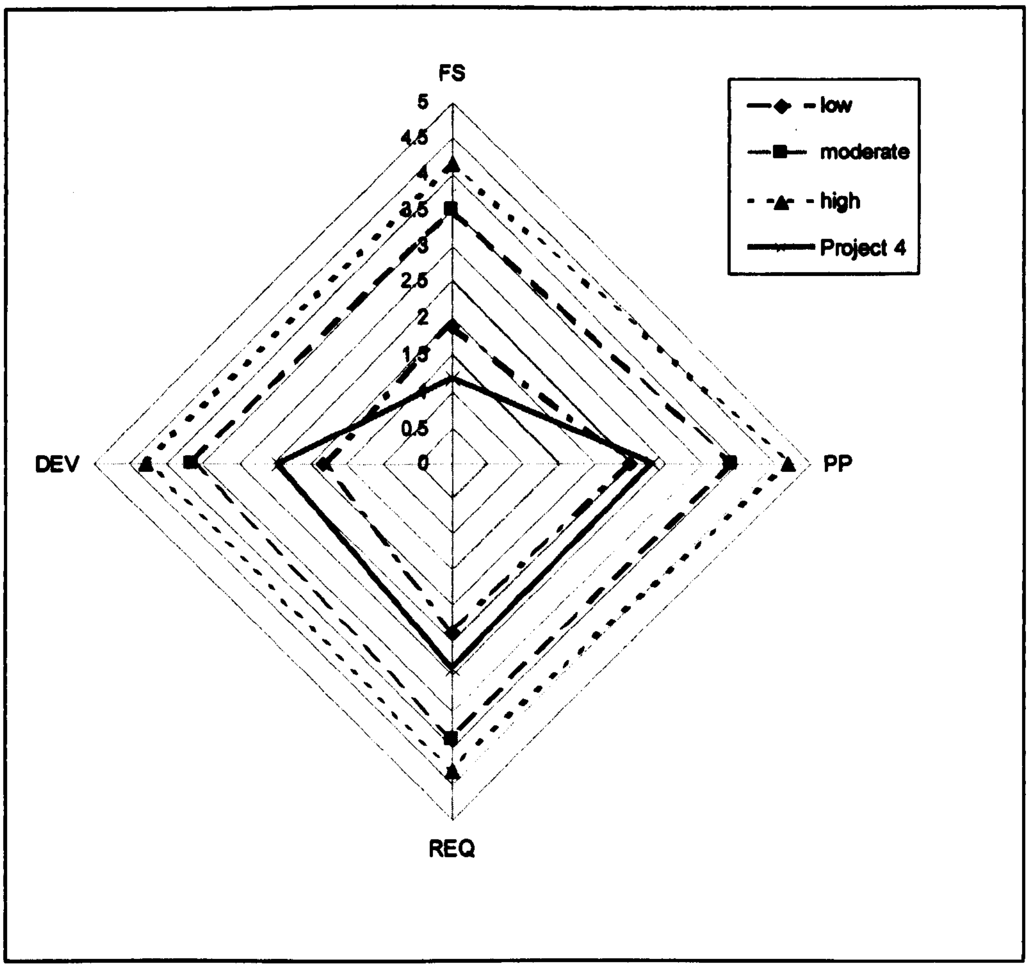


Figure 9.20 : The fuzzy model application for Project 4 (Impact of risk factor on cost overrun)

From the profiles of 'Project 4' in Figure 9.20, it can be seen that the level of risks for the feasibility study stage of 'Project 4' is very much lower than the 'low category' of the model. Where as, the risks level for the other 3 stages of 'Project 4' is higher than the 'low category' of the model, but lower than 'moderate category'.

Through this examples of 'Project 1 – Project 4', comparisons and differences can be made between the model's risk profile and the Project's risk profile, in order to assist the software practitioner in the decision making regarding the risk of software development project. The risk profiles can be used to prioritise, forecast or estimate the risk factors and focus on the most important and influential risk during the stages.

9.11. Summary

This chapter is an attempt to show that fuzzy theory application as an appropriate mechanism in dealing with the subjectivity of assessment of risk factors in the stages of the software development life cycle. Through this technique, the subjectivity is transferred to fuzzy membership function for easier comparisons and interpretations. The fuzzy computation of various combinations may assist IT practitioners and decision makers in formalising the types of thinking that are required in assessing the current risk environment and decision making process of their software development life cycle in a more simple and systematic manner than before.

CHAPTER 10 DISCUSSIONS

10.1. Introduction

The main objectives of the study were to identify the important risk factors for the likelihood occurrence and the risk factors that affect the cost overrun of the software development project, within the construct of project life cycle. The perceptions of the practitioners' views of the likelihood occurrence of risk factors and its impact on the cost overrun of the software project were discussed in the previous chapters, in the isolation of, the effectiveness of the risk mitigation strategies.

This chapter will discuss the significance and the relevancy of the research within the overall perspective of the software development project. Relationship and correlations of the risk occurrence, its impact on cost overrun and risk mitigation strategies were identified and highlighted.

10.2. The software risk constructs

From the available literature reviews, there are high number of conceptual frameworks, categorization, components and rankings, to explain different types of software development risk, risk management strategies and measures of software project performance. These frameworks produced an overlapping risk factors, components or categories.

The research focus, concerns the absence of an agreed risk framework for software projects over which there has been academic disagreement for some years (Wallace et al, 2004; Barki et al, 1993). The key issue appears to be the lack of a systematic framework by which the risk construct is developed and organized (Sherer & Alter, 2004a, 2004b). The objective within this research is to address this aspect by seeking to clarify the role of a project life cycle as the basis for the project risk construct.

The lack of organized framework could make it more difficult for managers and practitioners to identify, analyze the risk or even to suggest possible risk management strategies. It may also be difficult to identify the source or origin of the risk factors and tracking of the risk factors. The overlapping of many risk factors could make the risk management processes difficult and time consuming, especially as the risk management process also involved tracking and monitoring of the risk response strategies implemented. In addition, as most of the software development lifecycle process involved a lot of iterative process within the cycle, the risk management process can be a daunting task and complex.

In order to organize the risk factors in a structured framework, the research structured the risk factors based on the Project Management principle of managing a project. This is based on the fact that Project Management perspectives and principles are generally acceptable concept and widely used by most businesses and project managers for their projects. Furthermore, risk management is one of the knowledge areas in Project Management Body of Knowledge (PMBOK).

The basis of using the project management principle is that, categories, groupings and dimensions of risk factors based on project management perspectives could potentially provide a broader framing and hence be more widely applicable for thinking about what risks might be targeted for risk mitigation. Another reason for employing a project management perspective is that resulting risk categorisations will be placed in context of the stages, processes and activities within the software development project. As such, risk management becomes engaged with project management in a clearer manner.

The project management principles and perspectives were also quite easily understood and communicated by most practitioners, business managers or even other non-IT related staff within the organization. This could assist the team members to have a working understanding of risk factors within the context of their scope of work, and being more responsible and accountable for the consequences of their actions.

Furthermore, as most research suggested that risk identification and risk management process is also the responsibilities of other team members (and not just the project managers alone), this new framework could be a medium of enhancing communications between the team members in dealing with risk factors in software project.

10.3. Software risk factors classifications

The research reports an analysis of a survey of IT professionals with the objective of ascertaining their views concerning the occurrence of software risk factors, their impact on the cost overrun of software project, and possible mitigation of the risks involved in supporting the development of IT software projects.

Whilst there are many different, broad and overlapping definitions of project success and failures, completing the project within the estimated budget remains the project characterisation that is most frequently mentioned with respect to the reporting of project failure. Whichever way 'project failure' is characterized, it is hard to escape the idea of there being either a direct or indirect project cost implication. Moreover, it is distinguish in this survey and analysis between the risk occurrence and impact of risk factors on cost overrun as related to IT software project failure. Although the research impact measure referred to project cost overruns is not a comprehensive measure but does provide at least one metric which is firmly placed into a wider risk construct.

Furthermore, a cost consequence approach as this research would argue is most readily interpretable by survey participants (the research sample) and is one which naturally accords with a major characterization of project failure in popular perception. More specifically, the research contention is that a cost focus leads to survey responses that are directly relatable to the experience of the survey participant base, is more likely to be relevant to IT software project risk goals in a commonly understood manner and which, consequently, and is likely to enhance the psychometric properties of the research survey results.

The survey responses were analyzed using principal components analysis (PCA) to determine which of the risk factors cluster into statistically meaningful groupings. PCA has previously been used in grouping risk factors by previous researches. This initial part of the analysis reduces the candidate risk factor list to those most influential risk factors. The risk factors were then grouped as explanatory factor loadings into risk components. The aggregation of the remaining risk factors into risk components is a clustering process that is a PCA-determined method that forms cluster based on the degree of colinearity between risk factors.

The empirically derived risk components are then interpreted in terms which are meaningful in relation to the life-cycle of the project. This reversion to the research risk construct is an important element of the research contribution, on the basis of theory and further empirical work, why the risk components observed in practice are likely to be important generally. Thus, the research seek to interpret the risk components observed in relation to project life cycle which typically involves reconfiguring the observed components to those that meaningfully relate to the risk construct.

The survey participants were also asked to independently rank mitigation strategies in terms of their effectiveness as responses to IT software risk generally. The survey responses on a predetermined list of 30 generic strategies, chosen from the available literature.

The study analysis indicates that it is possible to identify a grouping of risk that is reflective of the different stages of the project life cycle. The findings suggest three identifiable clusters (Cluster 1-Feasibility study; Cluster 2-Project team management; Cluster 3-Technology requirement) when viewing risk from the likelihood of occurrence and three clusters (Cluster 1-Feasibility study; Cluster 2-Project team management; Cluster 3-Technology requirement) from a cost overrun perspective. The research account for this difference by suggesting that a more coherent framework, or risk construct, offered by viewing risk within the context of a project life cycle allows those involved in IT software projects to have a clearer view of the relationships between risk factors. It also allows the various risk components and the associated emergent clusters to be more readily identifiable. In this respect, the research believe to have contributed to the, as yet, unresolved debate concerning an appropriate risk construct in IT software projects.

10.4. Correlation of risk mitigation strategies and risk factors

In linking mitigation strategies to extracted components, the statistical approach is to estimate factor scores using Bartlett's method, and which are available as an option in standard SPSS packages. From this, will enable to correlate the mitigation rankings to both extracted components for risk likelihood and the impact of cost overruns to assess which risk mitigation strategies are likely to be used in practice. The analysis undertaken for the risk perceptions, the factor analysis, the clustering and risk mitigation strategies were all being discussed in isolation of one another. This part of the chapter will elaborate in greater details of interactions and correlation of the risk factors.

The detailed results of the correlation coefficient calculations between mitigation strategies and risk likelihood and the impact of cost overruns are reported in Table 10.1 and Table 10.2. Some additional methodological comments are pertinent at this point. Survey participant views were gathered concerning mitigation strategies without reference to their risk likelihood or impact on cost overrun: hence they were asked, from a list of commonly observed strategies reported in the academic and professional literature, to assess which strategies were most effective. These responses were then subjected to correlation analysis with the research extracted components to both risk likelihood and risk impact. In this way, the research are able to comment on both and also, crucially, any differences that emerge between them. This represents a criterion test of the research groupings because there are strong counter-factual possibilities that support potential falsification. First, it is possible that no significant correlations will be found. Assuming the strategic survey responses to be correct, this would falsify the extracted components and, by construction, the research groupings. Second, there may be some groupings for which there are no significant correlations. This would falsify (part of) the group composition if other groups exhibited significant correlations.

For reasons of clarity of presentation, only coefficients that are significantly different from zero at the 10% level or below are reported. Correlation coefficients between strategies (rows) and extracted risk components (columns) are reported in Table 10.1 and Table 10.2. Other extracted risk components shown in earlier chapters but not reported here, do not have any significant correlations. Similarly, only mitigation strategies with significant correlations are shown.

For risk likelihood Table 10.1, two main areas of strategic concern were observed in terms of the extracted components: both Project Implementation and Feasibility Study as extracted risk factors are correlated with at least 5 different risk mitigation strategies. The two largest correlations are, in fact, observed in relation to Project implementation risks. The corresponding strategies are concerned with Parallel or phased conversion (S19) with a correlation of 0.216 and with Staff training (S27) with a correlation of 0.202. It is not surprising that Project implementation and Feasibility appear as important areas of strategic activity for risk likelihood since they appear as 2 of the 3 clusters reported earlier in Chapter 7 for risk likelihood.

Specifically, the extracted components Feasibility study appeared in Cluster 1 and Project implementation appeared in Cluster 2. This link reinforces the earlier discussion in the previous chapters concerning the role of the build-up of risk. That is to say, poor feasibility planning embeds risk in the project at an early stage that manifest at later stages in the life cycle and it is at the project implementation stage where these failures crystallize. The research argued this point earlier that extracted Component 5 (Project implementation) related to the consequences of failure at the planning stage. Managers anticipate this in the subsequent analysis by putting in place strategic responses specifically directed at implementation problems. Seen together, it could be argued that the strategic emphasis on the extracted risk factors Feasibility study and Project implementation support the earlier findings that risk build-up is a feature of IT software development and that management is alert to the possibilities when implementation stages are reached.

Table 10.1: Correlations between mitigation strategies and the most likely risks to emerge

	Project user engagement	*Technology failure*	*Technology and system requirements*	*Project implementation*	*Feasibility study decision*
S1: Clear goals and objectives of the project				0.114	
S4: Comprehensive project planning and scheduling	0.120				0.134
S6: Identify critical and non-critical activities				0.114	
S7: Lessons learned from past software development projects		0.110			
S16: Prototyping	0.114	0.130		0.176	
S17: Timeframe for testing			0.117	0.148	
S18: User acceptance tests					0.155
S19: Parallel or phased conversion			0.131	0.216	
S20: Clear and detailed requirements					0.149
S21: Incorporating alternative development methodologies	0.150	0.125		0.177	
S23: Software security checklist and authentication processes		0.125			
S24: Cost control procedures					0.157
S25: Technical support teams				0.144	
S26: Resource planning					0.146
S27: Staff training				0.202	
S28: Effective lines of communication					0.146
S29: Effective project management and leadership	0.120				

Thus, in relation to the risks of making the wrong feasibility decision, there are significantly correlated strategies arising in relation to Comprehensive planning (S4), User acceptance tests (S18), Clear and detailed requirements (S20), Cost and control procedures (S24), Resource planning (S26), Staff training (S27), and Effective lines of communication (S28). These are combined with a range of 8 strategies that deal with Project implementation risks. Both feasibility study and project implementation project activities are resident in different stages of the project life cycle (Feasibility is the 1st out of 6 stages and Implementation is 5th as determined in the project life cycle description) and thus suggest distinct areas in the project life cycle where different strategies have different roles to play.

In fact, there are no instances where the same strategy is simultaneously correlated with both feasibility and implementation. This is suggestive, and at least not inconsistent, with the view that management consider different areas of the project life cycle as distinct strategically. What the research now observes is that the concern identified in terms of risk likelihood reported in Table 7.5 (in Chapter 7) is matched by the evidence of strategic response reported in Table 10.1. Finally, in terms of criterion validity, significant correlations were observed for each of the 3 Clusters and hence there is no strategic evidence to reject the groupings derived on the basis of content validity.

For the impact of the emerging risks on cost overrun in Table 10.2, it was noted that there is a fairly consistent view that strategies relating to incorporating alternative development methodology(ies) (S21), Software security checklist and authentication processes (S23), and Technical support teams (S25) are most likely to have a broad relevance over a range of extracted risk factors (they exhibit 3 or more significant correlations simultaneously over a range of extracted risk factors). In terms of their effectiveness as tools for risk reduction, they are therefore likely to exhibit relevance across a broad range of project activities. For the extracted risks themselves, the table produces an indication of which areas are a focus of attention for mitigation strategies.

As argued earlier, considerations of cost are at the forefront of reporting of failure of IT projects. In fact, the largest correlations observed relate to Prototyping strategies (S15) with a correlation of 0.210 and Parallel or phased implementation (S19) with a correlation of 0.196 and both correlate significantly to the Technology specification extracted risk. This perhaps gives an important steer towards the cautious, step by step management of implementation: that is, making sure that the technicalities of the new development are understood, work, and are robust to the environment they operate in. The lowest correlation is worthy of note, also. The correlation between strategies relating to Technical support teams and the extracted risk, Project team activities, is -0.013. Whilst negative, it is significant and suggestive of the fact that highly ranking strategies relating to Technical support are associated with lower ranking risks relating to the impact of cost overruns from Project team activities in a fairly consistent manner. They appear disjoint and may be reflective of the lack of observed congruence between the technical and non-technical areas.

Table 10.2: Correlations between mitigation strategies and risk components based on cost overrun impact

	Project team planning	Technology appropriateness	Technology specification	Technology and implementation	Project team activities
S1: Clear goals and objectives of the project			0.138		
S4: Comprehensive project planning and scheduling		0.125			
S6: Critical and non-critical activities			0.138		
S7: Lessons learned from past software development projects			0.131		
S8: Success criteria		0.120			
S12: External expertise/consultant					0.143
S13: Contingency plans			0.148		
S15: Prototyping			0.210		
S16: Analysis of development methodology					0.136
S17: Timeframe for testing			0.165	0.123	
S18: User acceptance tests					0.118
S19: Parallel or phased conversion			0.196	0.154	
S21: Incorporating alternative development methodology(ies)	0.111	0.119	0.168	0.133	
S22: System back-up					0.170
S23: Software security checklist and authentication processes		0.138	0.164	0.125	
S24: Cost control procedures					0.125
S25: Technical support teams			0.129	0.131	-0.013
S27: Staff training			0.170	0.141	

For the risk relating to the impact of cost overrun in Table 10.2, the research noted that, in terms of statistical significance, there are clear and multiple strategies employed in attempting to manage the risk relating to cost overrun arising in relation to Technology appropriateness, Technology specification, Technology and implementation, and Project team activities. The focus of attention on technical issues and the degree to which they attract strategic response is a new finding in addition to the usual reports concerning the importance of non-technical aspects of IT software projects. The explanation is that strategic thinking, when cost is the context, concentrates attention on technical failure. It is technical failure that is therefore apparently 'costly' in the minds of managers responsible for strategic responses and which provides an immediately interpretable metric of failure.

The research noted that an important change in terms of criterion validity for the groupings for cost overrun compared to risk identification. In cost overrun, the research does not observe any extracted components relating to feasibility and therefore reject the notion that a feasibility grouping has relevance for strategic responses when cost overrun is the context. The interpretation of this result is consistent with the earlier comments in Chapter 7 relating to risk build-up. That is to say, strategies correlating with cost overrun do not deal with feasibility risks. Considerations of cost and strategic response are therefore representative, the research would argue, of important relationships further down the project pipeline. Strategies are concerned therefore with implementation and post implementation issues and reinforce the notion of risks resident at different stages in the project cycle and also of changing importance of risk between risk identification and risk impact.

10.5. Fuzzy modellling

The model is based on fuzzy theory to reduce the influence of subjectivity and qualitative data in risk assessment. The usual qualitative assessments rely on scoring and weighting score. During risk assessment, practitioners usually express their ideas by assigning a rating to each identified risk by referring their own expressions. The model was developed based on the extracted risk factors and components in the factor analysis process in Chapter 7. Since software development project involve new technologies or new levels of knowledge, there bound to be a lot of uncertainties and unknown information, especially the risks, in particular during the early stages of the software development project

This fuzzy model is meant to represent imprecision, uncertainty and expressions or judgments that have no clear crisp values. The resulted risk factors was used to develop the membership function for the models, and illustrated in a spider net diagram, as explained in that chapter 9. The fuzzy models developed could be used on real life project for the purpose of testing the validity of the model. However, the testing of the model was only based on 'speculative projects' using a few of the 'respondents' from the sample.

The 3 clusters mentioned earlier with 45 risk factors, which were developed through factor analysis and data reduction, could be utilized to derive the score for each individual risk factors or stages for each projects. The results will enable software practitioners to identify the crucial risk factors for any particular project, and assign the appropriate risk mitigation strategies.

The example of the fuzzy model application was also highlighted in Chapter 9 (Fuzzy modellling). The fuzzy model application enables project managers, software practitioners or stakeholders to assess the risk factors and provide them with an illustration of the risk factors profile for the stages, based on the scores given, and take the appropriate actions.

10.6. Implications for research

The literature reviews highlighted that existing software risk frameworks and models have limited applicability and lack ability to easily communicate an organizing framework for software risk factors (Sherer & Alter, 2004a, 2004b). The previous researches had organized and categorized the risk according to dimensions, task, structure, element, attribute and other common characteristics of risk factors (Han & Huang, 2007; Costa et al, 2007; Wat & Ngai, 2005; Wallace et al, 2004; Schmidt et al, 2001; Ropponen & Lyytinen, 2000). This can be seen from the overlapping of risk factors and categories by previous researches. The focus of the previous researches was mainly on the characteristics of the risk factors but lacking of concentration on the project life cycles, processes and activities of the software project, which could make the management of the software project difficult and complex.

By organizing the risk factors into a general but adaptable model such as the project management life cycle approach could make the software risk factors more accessible and usable by managers. Project management life cycle approach is based on clearly defined concepts that are understandable, adaptable to a variety of contexts, and of practical use; since the project management perspectives are well known by most business managers. Unlike the framework and categories used by previous researches which were more specific and technically related, where it may not benefit the less knowledgeable business managers when it comes to software related risk factors.

One further important finding from this research compared to previous study, is the evidence of a perception of risk 'build-up' as the project proceeds through its life-cycle. In a number of areas from the survey analysis, it is possible to see evidence of the consequences of failure at the planning/feasibility stage being highlighted as important risk factors in the project and team management stage (which largely comes into play following the feasibility stage). This is previously unreported within the literature that risk-updating is an important project management exercise. This updating of risk enables project managers and IT professionals to at least consider the success or otherwise of earlier risk evaluations based on the evidence accumulating subsequently and this suggests that previous risk constructs may be too-static devices to capture the richness of the risk evaluation procedures in place.

The other findings compared to previous study is the evidence relating to the cost overrun view of risk that provided a stronger view of which components of risk were important, compared with risk likelihood. Moreover, the research on strategic response indicated different strategies as being effective between risk likelihood versus cost overrun. Of note, was the emergence of strategies specifically related to the risk of technical failure when considerations of cost overrun were asked-for. This is a new finding and it signifies clearly a management desire to offset technical failure as distinct from the risks relating only to non-technical activities that have been consistently reported as being at the forefront of management attention (Lopes & Flavell, 1998; Procaccino et al, 2005). The research did not see the results as being inconsistent with these findings, since the findings are concerned with the impact of cost overrun and, it is in relation to technical failure, that cost overrun is most likely to have a major impact.

Previous studies have frequently used project managers and occasionally users as their sample of study (Keil et al, 2008; Costa et al, 2007; Tesch et al, 2007; Wallace et al, 2004; Keil et al, 2002; Jiang & Klien, 2001; Boehm, 1991). But, this research used the main stakeholders and practitioners within the software development itself, which mainly consists of project managers, developers or programmers and IT support technical staff. The managing directors or board of directors perceptions was also perceived, as their perceptions were deemed to be relevant as the risk factors within the project could have a bearing on the running of their businesses as a whole.

10.7. Managerial implications

The models developed shows how the risk factors can be organized to make them more accessible and more easily communicated within the managers and project team. It provides practical approach that managers and project team can use for thinking about at whatever level of detail that is relevant and makes sense to them. This model's adaptability allows users, developers and managers to eliminate facets that are not important for their purposes. This model will truly be practical for use by business professionals and readily adaptable by users who may not be interested in all of the possible facets of other models.

Compilation of this information could provide means of assessing future projects. The measures developed here could then be used to create risk profile for each project. Potentially high risk projects could be flagged at an early stage so that appropriate decisions could be made about whether or not to continue with a high risk project, or to select an alternative course of action. Practitioners could also administer the instrument at multiple points during a project and track the changes in the riskiness of a project as it progresses from beginning to end.

In this research, the focus was on exploring the similarities and differences in how the software practitioners (project manager, managing directors, developers, IT staff) perceived software risk factors. By mapping these similarities and differences, the research have provided the practitioners with a more structured framework based on project management life cycle that encompasses the perceptions of the main software practitioners or stakeholders. Incorporation of these stakeholders perspectives on software risk factors are significant because focusing solely on any particular practitioner may result in some risk factors receiving a lower level of attention than they might actually deserve. To mitigate these software project risks, it is necessary to consider risk factors judged to be important by all groups and reconcile any differences. This should lead to a more comprehensive approach towards managing the risk associated with software projects.

The measures of software project risk developed in this study can also be used to learn more about the effectiveness of various risk mitigation tactics designed to reduce the severity of a risk factor's impact and to increase the likelihood of successful software development. In keeping with good project management practices, this would help to ensure that risk assessment is an ongoing process and not something that happens once at the outset of a project.

10.8. Summary

This study has described a few dimensions of software project risk that practitioners may use for identifying and managing the risks associated with software development projects. As different project participants viewed risk differently, the comparisons and differences in their assessments of the risk factors could provide insight on how to tackle the risks. The project life cycle approach adapted in this study would help to produce an understanding of how the risk profile of a project typically changes over time. By developing a more comprehensive list of risk factors, the research provide a basis of more comprehensive investigation that can be used in developing software project risk assessment guidelines.

The differences of risks perceptions based on the role taken within the development team were very important for the coordination among the various groups. This may indicate the need for improved communications in order to develop a shared understanding of project risk. Without a shared understanding of risk, it is unlikely that the Project Manager and their development team will be able to work together effectively, and there may be an increased potential for conflicts to arise.

CHAPTER 11 RESEARCH SUMMARY & CONTRIBUTIONS

11.1. Introduction

Although, there are a significant amount of research and literatures in this area, but most of the literatures were jumbled up and partially overlapping. Most of the research undertaken, did not really organise the risk factors identified in a more systematic manner and easily interpretable for software practitioners, project managers or business managers. Despite all these research, the software development project still suffers significant failures, albeit failures in terms of lack of quality, not meeting requirements, overrun the cost or even delay completion.

The customized software development life cycle adopted in this research based on the perspectives of Project Management Body of Knowledge (PMBOK), make up the research added new contribution the field. The stages of development life cycle used this research make it an easier and improved framework for managers, project managers, development team and users to follow through and understand the risk factors in software development project.

In this research, the main focus is on the occurrence of risk and the risk factors impact on cost overrun of the software project, as it is believed that, regardless whatever success criteria of a project, finishing the project within the budget is still main priorities for any businesses. The focus is on organising risk factors to make them more useful and meaningful for business managers, that helps them identify and mitigate these risks. The sheer numbers of risk factors makes it all more significant to use an organised framework. The fact that any software projects is also about bringing benefits to the business organization, this organised framework could also be used as medium of communication of risk factors between IS/IT personnel and other non-IT related business managers.

11.2. Summary

The significance of software is growing along with the progress of advanced technology and new level of knowledge. Every software development project faces a significant amount of uncertainty that is usually manifested as possible risk materialization. Generally, the success of a software development project is usually connected with the involved risks which mean project risks should be successfully mitigated in order to finish a software development project. There isn't a magic bullet to prevent all software projects from failure or to resolve all risks. However, it is still possible to mitigate some of the problems and increase the chances of success.

As explained in the previous chapters, many risk factors contributed the cost overrun of a software project. These factors can be technical, organisational, personnel or business oriented in nature. Although some of the individual risk factors may be more significant than the others, the project success usually depends on the combination of all risks, response strategies used to mitigate risks and a company's ability to manage them.

Risks can be identified and addressed in different phases of a software development project, but it is essential to identify risks as early as possible and address them promptly because the cost connected with exposed risks could be enormous. It is difficult to address all risks at the same time. In order to successfully address risks that arise on a software development project, it is necessary to divide risks into stages or more structured framework. Through the stages or framework, the relevancy and priority of the risk factors can easily be identified and addressed for the purpose of mitigating these risk factors. Furthermore, with this approach, risks will be mitigated in the early phases of software development, when the cost of a software development project is still small.

Throughout the research, most of the analysis and discussions tend to focus on the non-IT related issues of software development project. Although, there are important factors such as system requirements, development methodologies, testing or technical complexities, but most practitioners were in consensus that there were more crucial factors that have an impact on the cost overrun of a software project and effect on the overall success of project. Issues like project scope, resources, management support, user related issues, communication and other project management related factors were deemed critical by most practitioners.

Although it is certainly legitimate to reflect the concerns of software development practitioners and companies attempting to produce software to satisfy requirements, a risk literature that over-emphasizes these concerns inevitably under-emphasizes issues about systems and organizations which are subject to a broad range of risks more related to the work and the environment than to the software itself. This type of imbalance in the literature can lead to gaps in providing guidance for risk management. Focusing solely on IT/IS risk ignores the fact that IT systems or software are just one component of a manager's business environment and that many operational risks are due to the environment in which a software is operating rather than the software itself. Limiting the discussion to IS/IT risk can create a "responsibility gap" in an organization if IS/IT managers are responsible for managing IS/IT risk, and business managers, who should be identifying, assessing, and developing strategies for overall business risk, are left in the dark.

The results of the analysis discussed throughout this research, showed significant agreement among the practitioners for some factors, but there were also some disagreements. This shows that even experience practitioners can have different opinions of risk identification and mitigation. In a way, risk handling should not just be the responsibilities of project manager or one specific risk manager, but all parties involved. This means that every project member should identify and define risks connected with their problem area, and risks should also be identified and defined on the individual, team and organizational levels. And it is more easily understandable and less complex, to do this through the stages lifecycle as discussed in this research, as the interpretability of the stages mentioned is well understood by most practitioners and business managers.

This research approach was motivated in part by the need to improve communication between business and software professionals by using ideas and methods that are comfortable for business professionals in dealing with software risk management. The approach presented here focuses on the stages and framework of risk factors that is recognizable and understandable to business professionals, as the chances for success of a software development project are closely connected with successful risk addressing. The extensive listing of risks in literatures demonstrates the potential of organizing risks and risk factors in substantial detail using a model that managers can understand readily.

Enabling business and software practitioners to speak the same language supports enhanced communication that is necessary for collaboration between IT/IS and business professionals attempting to reduce IT-related business risks. Better ways of describing risk and relating it to everyday business projects and operations could help substantially.

11.3. Research Contributions

As explained in the early chapters, the extant research relates to the lack of agreed risk construct combined with an empirical validation of the links between risk likelihood, risk impact, and evidence of strategic response in terms of risk mitigation. The research contribution within its proposed risk construct and the evidence it seek to accumulate is designed to contribute in addressing these deficiencies. The survey then directly reports to key areas of risk identification, quantification and mitigation. The mitigation strategies reported are empirically those most likely to be observed as addressing the most likely risks and those most closely associated with cost overrun.

The research were also able to evidence opinion on risk likelihood, the impact of the risk of cost overrun, and the strategic responses that are likely to be effective in mitigating the risks that emerge in IT software projects. The contribution of the research relates to the assessment of risk within a construct that is defined in the context of a fairly broadly accepted view of the life cycle of projects. This research contribution is believed to be theoretically coherent and to link, in a consistent manner, risk identification and risk impact to risk response. The study was able to verify the effective mitigation strategies that are correlated to the risk components. In this way, the actions or consequences conditioned can be observed on identification of risk likelihood and risk impact on cost overrun as conceptualized in terms of a project life cycle, thereby 'closing the loop', as it were, of comprehensive risk management in relation to IT software projects.

The software risk construct based on the project management framework proposed in this research could facilitates a focus on roles and responsibilities, and allows for the coordination and integrations of activities for regular monitoring and aligning with the projects goals. This contribution would better enable management to identify and manage risk as they emerge with project stages and more closely reflect project activity and processes, and facilitate the risk management strategies exercise.

The research also developed a fuzzy theory based model to assist software practitioners in the software development lifecycle. This model could help the practitioners in the decision making process of dealing with risks in the software project. The employment of the model does not require any mathematical or complex algorithm knowledge. Only the subjective and qualitative assessment of the risk factors is needed.

Other main contribution is the evidence relating to the cost overrun view of risk provided a stronger view of which components of risk were important: compared with risk likelihood. The correlations between the risk factors (risk occurrence and impact on cost overrun) an the risk response strategies also indicated different strategies as being effective between risk likelihood versus cost overrun.

Other contributions of the research are presented as follows:-

i. Extraction of 104 risk factors for likelihood occurrence of risk and its impact on cost overrun of software project.
ii. Ranked risk factors for the likelihood of occurrence and its impact on cost overrun of software development project.
iii. Differences of perceptions of risk factors in software projects among the software practitioners.
iv. Grouping of 3 main clusters of risk factors through factor analysis and extraction of 45 most significant risk factors.
v. The research looks the risk factors in a scope of project management life cycle, which is more easily understandable and less complexity framework.
vi. Risk management criteria for the mitigating the software project risk.

11.4. Limitations of the research

It need to be point out that the risk construct based on project management life cycle is not exclusively drawn-up within the software practitioners in mind. The risk construct and the responses might vary dependent on which practitioner is asked. It is assumed that the risk construct proposed is accepted as a valid construct on the basis on limitations of existing risk construct from previous researches and good response rate received.

The responses received from the respondent were based on the respondents' previous experiences. But, whether their experiences were from their most recent projects, or from their overall judgement of their experiences with a number of previous projects, cannot be differentiated.

The research also did not analyse the differences of opinions of the practitioners with experience of different years or different number of projects. These limitations may create potential bias to the conclusion, as some practitioners may experience more bad projects than the good ones, or vice versa. Although the geographical locations of the respondents was not analysed in detail, but some respondents may have experiences of software projects in more than one country, and may also influence the responses given.

There's no certain way of knowing for certain that the responses received were from the practitioners themselves. There's a possibility that their assistants might be answering on their behalf. This research did not gave a very definitive interpretations of practitioners as to who is 'board of directors', 'project manager', 'developer', or 'IT support staff'. The research trust the practitioners to categorize themselves in what category they belongs to based on their own experiences and judgements.

The clustering of components into 3 main groups was within the judgement and interpretations of the researcher within the theoreotical concept of factor loading score of the components. However, other researchers may still interpretated the components differently and possibly produced a different results. The groupings and components of the risk factors may also subject to the risk construct or other life cycle based on the body of knowledge used by the researchers.

11.5. Recommendations for further research

In this current research, the study was limited to retrospectively assessing risk based on the experiences of practitioners, which is normally accounted from past projects. One potential area of further research involves using the constructs and measures described in this research to study how risk perceptions change during the course of a project, by using a real life case based scenario, and correlated the results with the project performance. Hence, the impact of the actual value and magnitude of the risk factors could be determined in terms of the cost overrun of the software project.

This research focused on the risk factors and its impact on the cost overrun of the software project, but it did not address any additional impact caused by the size of possible losses due to failure. Further research could investigate whether magnitude issues play into perceptions of risk. Since the model developed is based in simple mathematical computations, it could be developed into a computer programming and develop a software or prototype capable of assessing risk factors on different software projects. Other potential aspect for further research is by looking at the different perceptions of risk by different countries or continent into more detail. Even though the data for this is available in the responses of the questionnaires, but, due to the time constraints, this was not pursued in this study.

In practice, the risk management criteria or factors identified could also be applied on real case project and determined which strategies are suitable or ideal for any specific projects or project situations. The magnitude of the occurrence of the risk and its impact on cost overrun could also be established and differentiate, before and after, the mitigation strategies were applied. Furthermore, the impact of the risk mitigation strategies towards the project success could also be established.

The risk profile using the fuzzy modeling can be applied to real life project in practice to create risk profile for a project. Using this risk profile, decisions could be make on the potential impact of certain risk factors, and the high risk factors could be focused and highlighted early. This could assist the practitioners and decision makers, in making the appropriate decisions for the success of the software project.

REFERENCES

Aaen I. (2003). "Software process improvement: blueprints versus recipes"; IEEE Software 20(5): pp86–93.

Adeli, H. & Sarma, K. (2006). "Cost Optimization of Structures—Fuzzy Logic, Genetic Algorithms, and Parallel Computing"; John Wiley and Sons, West Sussex , United Kingdom.

Adler, T.R., Leonard, J.G., Nordgen R.K. (1999). "Improving risk management: Moving form risk elimination to risk avoidance"; Information & Software Technology; 41 (1); 1999; 29-34.

Ahmed A., Kayis B., Amornsawadwatana S., (2007). "A review of techniques for risk management in projects". *Benchmarking: An international Journal*; Vol 14; No.1; pp 22-36.

Ahn. J, Skudlark. A. (1997). " Resolving conflict of interest in the process of an information system implementation for advanced telecommunication services". *Journal of Information Technology*. Vol. 12. pp 3-13.

Alexander E.R.; E.A. Beimboro, C.R. Patton and L. Witzling. (1985). "Multi-objective decision making methods for transportation"; University of Wisconsin Milwaukee.

Ali Vakili-Ardebili; Abdel Halim Boussabaine. (2007). "Application of fuzzy techniques to develop an assessment framework for building design eco-drivers"; Building and Environment (42); pp 3785-3800.

Alter S & Sherer, S.A. (2004) "A General, but Readily Adaptable Model of Information Risk"; Communications of the Association for Information Systems 14, 1-28

Alter, S. (2001). "Which life cycle ... work system, information system, or software?" Communications of the Association for Information Systems, 7, Article 17.

Alter. S. (1992). *"Information Systems : A Management Perspective"*, Addison Wesley Publishing.

Alter. S. (2002), *"Information Systems : The Foundation of E-business"*, Pearson Education.

Altunok, E., Taha, M. M. R., Epp, D. S., Mayes, R. L. & Baca, T. J. (2006). "Damage pattern recognition for structural health monitoring using fuzzy similarity prescription"; Computer-Aided Civil and Infrastructure Engineering, 21(8), 549–60.

Ambrosini, V., Bowman, C., & Collier, N. (2009). "Dynamic Capabilities: An Exploration of How Firms Renew their Resource Base"; British Journal of Management, 20(s1): S9-S24

Atkinson, R. (1999) Project management: cost, time and quality, two best guesses and a phenomenon, its (sic) time to accept other success criteria. International Journal of Project Management , 17 (6), 337-342.
Available at http://in953.kelon.org/archives/in953/2004/papers/WhichLifeCycleCAIS7-17.pdf .
Last accessed 28 August 2009.

Avison D.E., Fitzgerald, G. (2003). "Where Now for Development Methodologies"; Communications of the ACM; January 2003; Vol 46; No.1; pp 79-82.

Avison, D., Fitzgerald, G. (2003). "Information System development :methodologies, techniques and tools"; McGraw Hill; 3rd edition.

Baccarini.D, Salm.G, Surrey.P. (2004). "Management of risks in Information Technology projects". *Journal of Industrial Management & Data Systems*; Vol 104; No.4; pp 286-295

Badri M.A, D. Davis and D. Davis. (2001). "A comprehensive 0–1 goal programming model for project selection"; *International Journal Project Management;* **19** ; pp. 243–252.

Bailey. A. (1998). " Uh-Oh. It's a Computer Systems Project......." *IEEE Engineering Management Review*. Winter. pp 21-25.

Balasubramaniam, P. Kulatilaka, N. Storck, J. (2000). "Managing IT investments using a real options approach". *Journal of Strategic Information Systems*. Vol.9, pp 39-62.

Ballantine, J & Stray, S. (1998). "Financial appraisal and the IS/IT investment decision making process"; Journal of Information Technology, 13 (1), 3-14.

Bannerman, P. L., (2008) "Risk and risk management in software projects: A reassessment"; Journal of Systems and Software. 81 (12), 2118-2133.

Bannerman, P.L. (2007). " Software project risk in the public sector". Proceedings of the 2007 Australian Software Engineering Conference; 10-13 April;; Melbourne; 2007; 389-398.

Bannerman, P.L. (2008). "Toward An integrated framework of software project threats". 19th Australian Conference on software engineering; IEEE 2008.

Bannermann, P. (2009). "Software development governance: A meta-management perspective"; International Conference on Software Engineering; Proceedings of the 2009 ICSE Workshop on Software Development Governance; Pages: 3-8

Barki H., Rivard S., Talbot J. (1993). "Toward as assessment of software development risk." Journal of Management Information Systems; 10(2); pp 203-223.

Barki, H., Rivard, S., Talbot, J. (2001). "An intergrated contingency model of software project risk management"; Journal of Management Information Systems; 17 (4); pp 37-69.

Barros, M., Werner, C., Travassos, G. (2004). "Supporting risks in software project management". *Journal of Systems & Software*; Vol.70; pp 21-35.

Beck, K. (1999). "Planning Extreme Programming"; Addison-Wesley; Longman Publishing.

Bellman. R.E., Zadeh, L.A. (1970). "Decision making in a fuzzy environment"; Management Science; 17 (4); pp141-175.

Benaroch, M. (2002). "Managing Information Technology Investment Risk: A Real Options Perspective"; Journal of Management Information Systems; Volume 19, Number 2 / Fall 2002.

Benediktsson, O., Dalcher, D., Helgi Thorbergsson (2006). "Comparison of Software Development Life Cycles: A Multiproject Experiment"; IEEE Proc.-Softw., Vol. 153, No. 3; pp 87-101.

Bennett, J.C., Bohoris, E.M., Aspinwall, R.C. (1996). "Risk analysis techniques and their application to software development"; European Journal of Operational Research; 95; 1996; 467-475

Beynon-Davis, P. (1995). "Information System Failure: the case of the London Ambulance Service's computer aided despatch project". *European Journal of Information Systems*; Vol.4; pp 171-84.

Block, R. (1983). The Politics of Projects (Yourdon Press, Prentice-Hall, Englewood Cliff, NJ).

Bodea, C.N., Dascalu, M.L. (2009). "Modelling research project risks with fuzzy maps"; Journal of Applied Quantitative Methods; Vol 4; No. 1; Spring 2009; pp 17-30.

Boehm, B. (1991). "Software risk management: principles and practices". IEEE Software; Vol. 8, No.1; pp. 32-41.

Boehm, B., Ross, R., (1989). "Theory-W software project management: principles and examples". IEEE Transactions on Software Engineering; Vol. 15; No.7; pp. 902-916.

Boehm, B., Turner, R. (2004). "Balancing agility and discipline: a guide for the perplexed"; Addison-Wesley; Boston.

Boehm, B.W. (1988). "A spiral model of software development and enhancement", IEEE Computer; pp. 26 – 37.

Boehm, B.W., T.E. Gray, T. Seewaldt. (1984) . "Prototyping vs. Specifying: A Multiproject Experiment"; IEEE Transactions on Software Engineering; SE-10(3): p. 290-303.

Boussabaine, A. H. and Elhag, T. (1999), "Applying fuzzy techniques to cash flow analysis"; Construction Management and Economics, Vol.17, No. 6, pp.745-755

Brandon, D. (2005). "Project Management for Modern Information Systems".Hershey, PA : USA; IRM Press 2005.

Buchanan. D (1991). "Figure ground reversal in systems development and implementation". in Human Jobs and Computer Interfaces. Nurminen.M, Weir.G (Eds). North Holland, Amsterdam, pp 213-216.

Burch, J.G., Grudnitski, G. (1983). "Information system: theory and practice; John Wiley and Sons.

Burns, R.B. (2000). Introduction to research methods. 4th edition. SAGE Publications.

Büyüközkan, G., Feyzioglu, G. (2004). "A fuzzy decision making approach for the new product development process under uncertainty"; International Journal of Production Economics 90 (1); pp27–45.

Capron. H, Perron. J.D. (1993). *"Computers and Information Systems : Tools for an information age"*, Benjamin Cummings Publishing.

Carter, B., Hancock, T., Morin, J., Robins, N. (1993). "Introducing RISKMAN: The European Project Risk Management Methodology; NCC Blackwell; Oxford.

Cerpa, N., Verner, J.M. (1996). "Prototyping: Some new results", Information Software Technology; Vol. 38; pp. 743 – 755.

Cervone. H.F. (2006). "Project risk management". OCLC Systems and Services: International digital library perspectives; Vol. 22; No.4; pp 256-262.

Chang S (2006). "An alternative methodology for Delphi-type research in IS key issues studies"; International Journal of Management and Enterprise Development,. 3(1/2): 147-168.

Chapman, C.B., Cooper, D.F., (1983). "Risk analysis: testing some prejudices". European Journal of Operational Research; Vol. 14; pp.238-247.

Chapman, R.J. (1998). "Effectiveness of working group risk identification and assessment techniques". *International Journal of Project Management;* Vol. 16; No. 6; pp 333-343.

Charette, R.N. (1989). Software engineering risk analysis and management. McGram-Hill, New York.

Charette, R.N. (2005). " Why Software fails"; IEEE Spectrum; 42 (9); 2005; 42-49.

Chen Y.W., G.H. Tzeng. (2001). "Using fuzzy integral for evaluation subjectively perceived travel costs in a traffic assignment model,; European Journal Operational Research; **130** (3); pp. 653–664.

Chen, K., Gorla, N. (1998). "Information system project selection using fuzzy logic"; IEEE Transaction System Management Cybernetic – Systems Human; 28 (6); pp849-855.

Chen, S.J., Hang, C.L. (1992). "Fuzzy Multiple Attribute Decision-Making: Methods and Applications; Springer-Verlag, Berlin.

Chen-Tung Chen, Hui-Ling Cheng (2009). "A comprehensive model for selecting information system project under fuzzy environment"; International Journal of Project Management; 27; pp389-399.

Civanlar, M.R. and Trussell, H.J. (1986), "Constructing membership functions using statistical data"; Fuzzy Sets and Systems, 18, pp. 1-13

Clegg, C., Warr, P., Green, T., Monk, A., Allison, G., Lansdale, M. (1989). "People and Computers: How to evaluate your company's new technology". Ellis Horwood, Chichester.

Clegg. C, Axtell. C, Damadoran. L, Farbey. B, Hull. R, Tomlnson. C, (1997). "Information Technology: A study on performance and the role of human and organisational factors". *Ergonomics*, Vol. 40. No. 9, pp 851-871.

Clifton, H.D. (1990). "Business data systems: a practical guide to system analysis and data processing". Prentice Hall.

Cockburn, A. (2001). "Agile software development"; Addison-Wesley; Longman Publishing.

Coles, R. & Hodgkinson, G.P. 2008. "A psychometric study of information technology risks in the workplace"; Risk Analysis, 28, 81-93.

Collingridge, D. 1984. Technology in the policy process - the control of nuclear power. London:

Collingridge, D. 1992. The management of scale: Big organizations, big decisions, big mistakes. London: Routledge.

Comrey, A.L, H. B. Lee. (1992). "A first course in factor analysis"; 2nd edition; Hillside, NJ: Erlbaum.

Conger, S., Karen D. Loch, B. Loerine Helft. (1995). "Ethics and information technology use: a factor analysis of attitudes to computer use"; Information Systems Journal; Vol 5; Issue 3, pp 161-183

Conrow, E.H., Shishido, P.S. (1997). " Implementing risk management on software intensive projects"; IEEE Software; 14 (3); 1997; 83-89.

Costa, H.R., Barros, M.O., Travassos, G.H. (2007). " Evaluating software project portfolio risks". Journal of Systems & Software; 80 (1); 16-31.

Czuchny, A.J. Yasin, M.M. (2003). "Managing the project management process". *Journal of Industrial Management & Data Systems*; Vol.103; No.1; pp39-46.
Francis Pinter.

Dansereau, F. & F. J. Yammarino. 2002. Overview: The many faces of multi-level issues. In The many faces of multi-level issues, eds. F. J. Yammarino & F. Dansereau, xiii-xix. Oxford: JAI (Elsevier Science Ltd).

Darbra, R.M., Eljarrat, E., Barcelo, D. (2008). " How to measure uncertainties in environmental risk assessment"; Trends in Analytical Chemistry; Vol 27; No. 4; pp 378-385.

Davis A.M., E.H. Bersoff, &. Comer, E.R (1998) "A Strategy for Comparing Alternative Software Development Life Cycle Models"; IEEE Transactions on Software Engineering, 14, 10, 1453-1461.

Dey P.K., Kinch J., Ogunlana S., (2007. "Managing risk in software development projects : case study". *Industrial Management and Data Systems;* Vol 107; No.2; pp284-303.

Dikmen, I., M. Talat Birgonul, Sedat Han. (2007). "Using fuzzy risk assessment to rate cost overrun risk in international construction projects"; International Journal of Project Management 25; pp.494-505.

DiStefano C., M Zhu & D. Mindrila, 2009. Understanding and using factor scores: considerations for the applied researcher. Practical Assessment, Research and Evaluation, 14, 20.
Available online at http://pareonline.net/pdf/v14n20.pdf Last accessed, 27 July 2010.

Doherty. N.F, King. M. (1998). " The importance of organisational issues in systems development". *Journal of Information Technology & People*. Vol.11. No.2. pp 104-123

Dong, W. and Wong, F. (1985). "Fuzzy computation and risk in decision analysis"; Civil Engineering systems, 2, pp.201-8

Down A., M. Coleman and P. Absolon. (1994). "*Risk Management for Software Projects*, McGraw-Hill, New York.

Drew, S. (2005). " Reducing enterprise risk with effective threat management"; Information Security Management; 13 (6); 2005; 37-42.

Dubois, D. and Prade, H. (1980), Fuzzy Sets and Systems: Theory and Application, Academic Press, New York.

Easterby-Smith, M., Lyles, M. A., & Peteraf, M. A. (2009). "Dynamic Capabilities: Current Debates and Future Directions"; British Journal of Management, 20(s1): S1-S8.

Edwards. C, Ward. J, Bytheway. A, (1995). " *The Essence of Information Systems* ". 2nd Edition. Prentice Hall. 1995.

Eloff, J.H.P., Labuschangne, L., Badenhorst, K.P. (1993). "A comparative method for risk analysis methods". Computers and Security; Vol.12; No. 6; pp 597-603.

Engel, A., Barad, M. (2003). "A methodology for modeling VVT risks and costs"; Systems Engineering Journal 6 (3); pp135–151.

Engel, A., Last, M. (2007). "Modeling software testing costs and risks using fuzzy logic paradigm". Journal of Systems and Software 80 (6); pp817–835.

Engel A., Shacher S. (2006). "Measuring and optimising systems' quality costs and project duration; System Engineering Journal; 9 (3); pp 259-280.

Epich, R., Person, J., (1994). "A fire drill for business". Information Strategy: The Executive's Journal; pp. 44-47.

Eun Hee Kim and Yongtae Park, (2006). " An exploratory study of risks in information system development projects : using association rule mining". International Journal of Technology

Ewusi-Mensah, & Przasnyski, K. (1991). "Information systems project abandonment: an exploratory study of organizational practice"; MIS Quarterly; 15(1), 67-88.

Ewusi-Mensah, & Przasnyski, K. (1994). "Factors contributing to the abandonment of information systems development projects"; Journal of Information Technology, 9, 185-201

Ewusi-Mensah, K. (1997). "Critical issues in abandoned information systems development projects"; Communications of the ACM, 40(9), 74–80.

Fairley, R., (1994). "Risk management for software projects". *IEEE Software;* Vol.11; No.3; pp 57-67.

Fitzgerald, B. (1998). "An empirical investigation into the adoption of system development methodologies"; Information and Management; 34; pp317-328

Fenton, N., Krause, P., Neil, M. (2002). " Software measurement: Uncertainty and causal modelling"; IEEE Software; 19 (4); 2002; 116-122.

Field, A. (2005). Discovering statistics using SPSS. 2nd edition. SAGE Publications.

Gemmer, A. (1997). " Risk Management: moving beyond process"; Computer; Vol 30; No.5; 33-41.

Ghotb, F., Warren, L., (1995). "A case study comparison of the analytic hierarchy process and fuzzy decision methodology"; Engineering Economic; 40 (3); pp233-247.

Glass, R.L. (2001). " Frequently forgotten fundamental facts about software engineering"; IEEE Software; Vol 18; No. 3; 110-112.

Glass, R.L., (1999). "Evolving a new theory of project success." Communications of the ACM; Vol. 42; No.17.

Goldstein, H. (2005). "Who killed the virtual case file?" *IEEE Spectrum.* 24-35.

Gonzalez, B., Adenso-Diaz, B., Gonzalez-Torre, P.I.(2002). "Resource Conservation"; Recycling; 37 (1); p61.

Guadagnoli E & Velicer, W. F. (1988) "Relation of sample size to the stability of component patterns"; Psycholigical Bulletin, 103(2), 265-275.

Halliday, S. Badenhorst, K. Solms, R. (1996). "A business approach to effective IT risk analysis and management". *Information Management and Computer Security.* Vol.4. No.1. pp 19-31.

Han, S. H. & Diekmann, J. (2004). " Judgment-based cross-impact method for predicting cost variance for highly uncertain projects"; Journal of Construction Research, 5(2), 171–92.

Han, W. M. & Huang, S-J. (2007). "Empirical analysis of risk components of software projects"; Journal of Systems and Software, 80 (1), 42-50.

Handy, C. 1994. The empty raincoat: making sense of the future. London: Hutchinson.

Handy, C. 1995. The age of unreason. New thinking for a new world. London: Random House.

Harter, D., M. Krishnan, Slaughter, S. (2000). "Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development"; Management Science; Vol. 46; No. 4; pp. 451-466.

Hedelin, L., Allwood, C.M. (2002). "IT and strategic decision making". Industrial Management & Data Systems; Vol. 102; No.3; pp 125-139.

Heemstra, F. J., & Kusters, R. J. (1996). "Dealing with risk: A practical approach"; Journal of Information Technology, 11(4), 333–346.

Heldman, K. (2004). PMP – Project Management Professional Study Guide. Sybex Incorporated. Alameda USA. (E-Books)

Henderson-Sellers, B & Edwards, J. M. (1990) . "The object-orientated systems life cycle"; Communications of the ACM, 33 (9), 142– 159.

Herbsleb, J., A. Carleton, J. Rozum, J. Siegel, D. Zubrow. (1994). "Benefits of CMM-Based Software Process Improvement: Initial Results," Technical Reports CMU/SEI-94-TR-013 and ESC-TR-94-013; Software Engineering Institute; Carnegie Mellon Univ.

Herrera, F., Herrera-Viedma E. (2000). "Linguistic decision analysis: steps for solving decision problems under linguistic information"; Fuzzy Set System; 115; pp67-82.

Hinde, S. (2005). " Why do so many majot IT projects fail". Coumputer Fraud & Security; January 2005,

Hornby, C. Clegg, C. Robson, J., McClaren, C., Richardson, S., O'Brien, P., (1992). "Human and organizational issues in information systems development". Behaviour and Information technology, Vol.11, No.3, pp160-174.

Huang, S., Chang, I., Li, S., Lin, M. (2004). "Assessing risks in ERP projects: identify and prioritise the factors". Industrial Management & Data Systems; Vol. 104; No. 8; pp. 681-688.

Huang, S.J., Han, W.M. (2008). "Exploring the relationship between software project duration and risk exposure; a cluster analysis". Information and management; 45; 2008; 175-182.

Huisman M., Iivari, J. (2006). “Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers”; Information and Management; 43; pp29-49.

Hussain. K.M, Hussain. D.S. (1995), *“Information Systems for business”*, Prentice Hall.

Hutcheson, G., Sofroniou, N. (999). The multivariate social scientist. SAGE Publications. London.

Ibbs, C.W., Kwak, Y.H., “Assessing project management maturity”; Project Management Journal; 31 (1); 2000; 32-43.

Iivari.J., (1996). "Why are CASE tools not used?”; Communication of ACM; Vol. 39; No.10; pp. 94 – 103.

Inchusta, Sanchez. P.J, Pina. A, Buldain. G (1998). “Economic Feasibility and success factors in IT implementation”. IEEE International Engineering Management Conference. 1998. pp 88-93. Intelligence and Planning; Vol. 2; No. 4.

Iversen, J., Mathiassen, L., Neilsen, P.A. (2004). “Managing risks in software process improvement: an action research approach”; MIS Quarterly; 28 (3); 395-433.

Jackson, S.L. (2006). Research Methods & Statistics: A critical thinking approach. 2nd edition. Thomson Wadsworth Publications.

Jackson, W. (2002) Poverty and Agricultural Policies: We Ain't Winnin' Because the Old Dominant Idea Has a Way of Reasserting Itself, Population and Environment, 24(1), pp. 55-67.

Jeganathan, C. (2003) Development of Fuzzy Logic Architecture to assess the sustainability of the Forest Management, International Institute for Geo-Information Science and Earth Observation, ENSCHEDE, The Netherlands, Master degree Thesis.

Jiang J., Klein G., (2000). “Software development risks to project effectiveness”. *Journal of Systems & Software*; Vol.52; pp 3-10.

Jiang, J., Klein, G., Discenza, R. (2001). “ Information system success as impacted by risks and development strategies”; IEEE Transactions on Engineering Management; 48 (1); 46-55.

Jiang J., Klein G., Chen H.G., Lin L., (2002). “ Reducing user related risks during and prior to system development”. *International Journal of Project Management*; Vol 20; pp507-515.

Jiang, J., Klein, G. Ellis, T.S. (2002). “A measure of software development risk”; Project Management Journal; 33 (3); 2002; 30-41.

Jin, X.H., Doloi, H. (2009). “Modelling risk allocation in Privately financed Infrastructure projects using Fuzzy logic”; Computer Aided Civil and Infrastructure Engineering; 24; pp 509-524.

Kaiser, H.F. (1960). The application of electronic computers to factor analysis. Educational and Psychological Measurement, 20 (1), 141-151.

Kaiser, H.F. (1974). An index of factorial simplicity. Psychometrika, 39; 31-36.

Kass, R.A. & Tinsley, H.E.A. (1979) Factor Analysis, Journal of Leisure Research, 11, 120-138.

Kearney. A.T. (1990). " Barriers to the successful application of information technology". DTI & CIMA. London.

Keil, M, Tiwana, A, &. Bush, A. (2002). Reconciling user and project manager perceptions of IT project risk: a Delphi Study. Information Systems Journal, 12 (2), 103–119.

Keil, M. Mann, J. Rai, A. (2000). "Why software projects escalate : An empirical analysis and test of four theoretical models". *MIS Quarterly*, Vol. 24 (4), pp 631-664.

Keil, M., Cule, P.E., Lyytinen, K., Schmidt, R.C. (1998). "A framework for identifying software project risks". Communications of the ACM; Vol. 41; No. 11; pp 76-83.

Keil, M., L., Wallace, D. Turk, G. Dixon-Randall & Nulden, U. (2000). An investigation of risk perception and risk propensity on the decision to continue a software development project. Journal of Systems and Software. 53 (2), 145-157.

Keil, M., Li, L., Mathiassen, L., Guangzhi, Z. (2008). " The influence of checklists and roles on software practitioner risk perception and decision making"; Journal of Systems and Software; 81; 908-919.

Keil, M., Mann, J. (1997). The nature and extent of IT project escalation: results from a survey of IS audit and control professional; IS Audit control journal; 1, 40-48.

Kerzner, H. (2003). "Project management: a systems approach to planning, scheduling and controlling; 8th edition; John Wiley & Sons.

Khalifa, M., Verner J.M. (2000). "Drivers for software development method usage"; IEEE Transactions on Engineering Management; Vol 47; No.3.

Kingston. J. (2004). " Conducting Feasibility Studies for Knowledge Based Systems". *Journal of Knowledge Based System*. Vol. 17 (2004), pp 157-164

Koen Milis, Roger Mercken. (2002). "Success factors regarding the implementation of ICT investment projects; International Journal Production Economics; 80; pp105-117.

Kosko, B., 1996. Fuzzy Engineering. Prentice Hall. ASIN: 0131249916.

KPMG (2005). Global IT project management survey. KPMG. Australia.

Kumar, R. L., (2002). Managing risks in IT projects: an options perspective. Information and Management, 40 (1), 53-74.

Kuruppuarachchi, P R., Mandal, P., & R Smith, (2002) "IT project implementation strategies for effective changes: a critical review", Logistics Information Management, Vol. 15: 2, pp.126 - 137

Kwak Y.H., Stoddard. J., (2004) "Project risk management: lessons learned from software development environment." *Technovation;* Vol 24; pp915-920.

Kwok Tai Hui & Biau Liu. (2004). " A Bayesian Belief Network Model and tool to Evaluate Risk and Impact in Software development projects"; IEEE.

Larman, C & Basili, V. R. (2003). Iterative and Incremental Development: A Brief History. IEEE Computer (IEEE Computer Society), 36 (6), 47–56.

Laudon, K.C., Laudon, J.P. (1995). "Essentials of management information systems". Prentice Hall.

Lee, H-M. (1999). "Generalization of the group decision making using fuzzy sets theory for evaluating the rate of aggregative risk in software development"; Information Sciences 113 (3–4); pp301–311.

Lee, H-M., Lee, T-Y., Chen, J.J. (2003). "A new algorithm for applying fuzzy set theory to evaluate the rate of aggregative risk in software development"; Information Sciences 153; pp177–197.

Leslie Willcocks, Catherine Griffiths. (1994). "Predicting risk of failure in large-scale Information Technology projects; Technological forecasting and social change; Vol 47; issue 2; pp205-228.

Li, J., Conradi, R., Slyngstad, O.P.N., Torchiano, M., Morisio, M., Bunse, C. (2008). " A state of the practice survey of risk management in development with off the shelf software components"; IEEE transactions on software engineering; Vol 34; No.2; March/April 2008.

Linberg, K.R. (1999). "Software developer perceptions about software project failure: a case Study"; Journal of System and Software, 49 (2), 177-192.

Liu, S, Zhang, J, Keil, M & Chen, T. (2009). "Comparing senior executive and project manager perceptions of IT project risk: a Chinese Delphi study"; Information Systems Journal, pages 1-37. Article viewed online before printed publication.

Lopes , M., & Flavell., M, (1998). "Project appraisal—a framework to assess non-financial aspects of projects during the project life cycle." International Journal of Project Management, 16 (4), 223-233.

Lynne. M, Benjamin.R (1997). " The Magic Bullet Theory in IT-enabled Implementation". *Sloan Management Review*. Vol. 38. No. 2. pp 55-68.

Lyytinen, K., Mathiassen, L., Ropponen, J. (1996). " A framework for software risk management"; Journal of Information Technology; 11 (4); 1996; 275-285.

Lyytinen, K., Mathiassen, L., Ropponen, J. (1998). " Attention shaping and software risk: A categorical analysis of four classical risk management approaches"; Information Systems Research; 9 (3); 1998; 233-255.

Macher, J. T., & Mowery, D. C. (2009). "Measuring Dynamic Capabilities: Practices and Performance in Semiconductor Manufacturing"; British Journal of Management, 20(s1): S41-S62.

Maccallum, R.C., Widaman, K.F., Zhang, S., Hong, S. (1999). "Sample size in factor analysis"; Psychological Methods; 4(1); 84-99.

Mahaney, R. C. & Lederer, A. L. (2010). "The role of monitoring and shirking in information systems project management"; International Journal of Project Management, 28, Issue 1, January 2010, Pages 14-25

March, J.G., Shapira, Z. (1987). "Managerial perspectives on risk and risk taking"; Management Science; 33 (11); pp 1404-1418.

Martin J., Odell, J.J. (1992). "Object-Oriented Analysis and Design". Prentice-Hall.

Martin. C, Powell. P (1992). *"Information Systems : A management perspective"*, McGraw Hill.

Martin. E.W, Brown. C, Hayes. D, Hoffer. J, Perkins. W. (2002). " *Managing Information Technology*". 4th Edition. Prentice Hall 2002.

Mata, F.J., Fuerst, W.L. & Barney, J.B. (1995) "Information technology and sustained competitive advantage: a resource-based analysis"; MIS Quarterly, 19, 487–505.

Mathiassen, L., T. Seewaldt, J. Stage.(1995). "Prototyping vs. Specifying: Principles and Practices of a Mixed Approach"; Scandinavian Journal of Information Systems; 7(1): p. 55-72

Maylor, H. (2003). Project Management (3rd edition). Pearson Education Limited.

McFarlan, F.W. (1981). Portfolio approach to information systems. Harvard Business Review, 59 (5), 142–150.

McKone, T.E., and Deshpande, A.W. (2005). "Can Fuzzy Logic Bring Complex Environmental Problems into Focus?"; Environ. Sci. Technology; 39 (2); pp 42A–47A.

Melville, N., Kraemer, K. & Gurbuxani, V. (2004) Review: IT and organizational performance: an integrative model of IT business value. MIS Quarterly, 28 (2), 283–322.

Meredith, J.R., Mantel, S. (2006). Project Management : a managerial approach (6th edition). John Wiley & Sons Inc.

Miller, R.L., Acton, C., Fullerton, D.A., Maltby, J. (2002). SPSS for social scientists. Palgrave Macmillan.

Mobey. A, Parker. D (2002). " Risk Evaluation and Its Importance to Project Implementation". *Work Study*. Vol 51. No. 4. (2002). Pp 202-206.

Morgan, G.A., Leech, N.L., Gloeckner, G.W., Barret, K.C. (2004). SPSS for introductory statistics: Use and interpretation. 2nd edition. Lawrence Erlbaum Associate Publishers.

Mykytyn, P.P, Mykytyn, K. Bandyopadhyay, (1999), "A framework for integrated risk management in IT". *Management Decision;* Vol.37 No.5, pp 437-444. MCB University Press.

Na, K.S., Simpson, J.T., Li, X., Singht, T., Kim, K-Y. (2007). "Software development risk and project performance measurement: Evidence in Korea"; Journal of Systems & Software; 80 (4); 596-605.

Neumann, Donald E. (2002). "An enhanced Neural Network Technique for software risk analysis"; IEEE Transactions on Software engineering; Vol 28, No. 9; September 2002; pp 904-912.

Nait-Said, R., F. Zidani, N. Ouzraoui. (2008). "Modified risk graph method using fuzzy rule-based approach", Journal of Hazardous Materials; 2008.

Naquin, C.E., Tynan, R.O., (2003). "The team halo effect : why teams are not balmed for their failures." Journal of Applied Psychology; Vol. 88; pp 332-340.

Ngai, E.W.T., Wat, F.K.T., (2005). " Fuzzy decision support system for risk analysis in e-commerce development"; Decision Support systems; 40 (2); 235-255

Nidumolu, S., (1996). "A comparison of the structural contingency and risk-based perspectives on coordination in software development projects". *Journal of Management Information Systems*; Vol.13; No.2; pp 77-113.

Osmundson, J.S., James B., Martin J. Machniak, Mary A. Grossman. (2003). "Quality management metrics for software development"; Information & Management; Volume 40; Issue 8; pp 799-812.

Perera, M., Ranasinghe, M. (2006). "Prompt list for risk management in Sri Lanka"; International Conference (ICIA); IEEE.

Pinto, J.K. (1998). "Project Management handbook; The project management institute; 1st edition; Jossey-Bass; San Francisco; CA.

Palomo, J., D. R. Insua & F. Ruggeri (2007) Modeling External Risks in Project Management. Risk Analysis, 27, 961-978.

Palvia. P, Palvia. S. (1988). " The Feasibility Study in Information Systems : An analysis of criteria and contents". *Journal of Information Management*. Vol.14 (1988). Pp 211-224

Pender, S. (2001). " Managing Incomplete Knowledge: Why risk management is not sufficient"; International Journal of Project Management; 19 (2); 2001; 79-87.

Perrow, C. 1984. Normal Accidents. New York: Basic Books.

Pfleeger, S.L. (2000). " Risky business: What we have yet to learn about risk management"; Journal of System and Software; 53 (3); 2000; 265-273.

Phillis Y. A. and Andriantiatsaholiniaina, Luc. A. (2001), Ecological Economics, vol. 37, issue 3, pp. 435-456

PMI (2004), A guide to the Project Management Body of Knowledge, Project Management Institute, 3rd ed.

PMP (2006). PMP in depth : Project Management Professional Study Guide for PMP & CAPM exams. Course Technology Incorporated. Boston. USA. (E-Books)

Pritchard, C.L. (1997). Risk Management – Concepts and Guidance; ESI International, Arlington.

Procaccino, J.D. Verner, J.M. (2002). " Software practitioner's perception of project success : a pilot study". International Journal of Computers, the Internet & Management; Vol 10 (1); pp 01-02.

Procaccino, J.D.,, June Verner, Scott P.Overmyer, Marvin E.Darter (2002). "Case study : Factors for early prediction of software development success". *Journal of Information & Software Technology*; Vol.44; pp 53-62.

Procaccino, J.D. Verner, J.M. Shelfer, K.M. Gefen, D. (2005). " What do software practitioner's really think about project success : an exploratory study" Journal of Systems & Software; Vol 78 (2); pp 194-203

Project Management Institute (PMI). (2000). *Project Management Body of Knowledge (PMBOK).*

Punch, K.F. (2005). Introduction to Social Research: Quantitative & qualitative approaches. 2nd edition. SAGE Publications.

Rainer, R.K.J.R., Snyder, C.A., Carr, H.H. (1991). "Risk analysis for information technology"; Journal of Management Information Systems; 8 (1); 1991; 129-147.

Reason, J. T. 1997. Managing the risks of organizational accidents. Aldershot: Ashgate.

Reel, J.S. (1999). "Critical Succes factors in software projects"; IEEE Software; 16 (3). 18-23.

Remenyi, D. (1999). "Stop IT project failures through risk management". *Computer Weekly Series*; Butterworth Heinemann; Oxford.

Riemenschneider, C.K., Bill C. Hardgrave, Fred D. Davis. (2002). "Explaining Software Developer Acceptance of Methodologies: A Comparison of Five Theoretical Models"; IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 28, NO. 12.

Roisenberg, M., Cintia Schoeninger, Reneu Rodrigues da Silva. (2008). "A hybrid fuzzy-probabilistic system for risk analysis in petroleum exploration prospects"; Expert Systems with Applications; 2008.

Rietveld, T. & R. van Hout, 1993. Statistical Techniques for the Study of Language and Language Behaviour. Mouton de Gruyter.

Ropponen, J., & Lyytinen, K., (2000). "Components of software development risk: How to address them? A project manager survey"; IEEE Transactions on Software Engineering; 26 (2), 98-112.

Royce, W (1970), Managing the Development of Large Software Systems, Proceedings of IEEE WESCON 26 August: 1-9. Available at http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf .Last accessed 28 August 2009.

Rubinstein, D. (2007) "Standish Group report: there's less development chaos today"; Software Development Times, 169, 1–2.

Santhanam R., J. Kyparisis. (1995). " A multiple criteria decision model for information system project selection"; Computational Operational Research; **22** (8); pp. 807–818.

Schach, S.R. (1996). "Classical and Object-Oriented Software Engineering"; Irwin .

Schniederjans M.J., K. Flower,. (1989). "Strategic acquisition analysis: a multi-objective approach"; Journal Operational Research Society; **40** (4); pp. 333–345.

Schniederjans M.J., R. Santhanam, (1993). "A multi-objective constrained resource information system project selection problem"; *European Journal Operational Research;* **70**; pp. 244–253.

Schwaber, K. (2001). "Agile Software Development with SCRUM"; Prentice Hall.

Senn, J.A. (1995). "Information Technology in business: principles, practices, opportunities". Prentice Hall.

Shenhar, A.J., Dvir, D. (2007). "Project management research: the challenge and opportunity; Project management journal; 38 (2); pp93-99.

Shull, N.D.P. (2006). "Project evaluation using fuzzy logic and risk analysis techniques"; Masters Thesis; MSC in Industrial Engineering; University of Puerto Rico.

Simister, S.J. (2004). "Qualitative and quantitative risk management; in Morris, P.W.G., Pinto, J.K., (Eds); The Wiley Guide to Managing project; John Wiley and Sons; Hokoben; pp 30-47.

Smith, H. J., Milberg, S. J., & Burke, S. J. (1996). "Information privacy: Measuring individuals' concerns about organizational practices"; MIS Quarterly; 20(2); pp167–196.

Subramaniam, G.H; Jiang, J.J., Klien, G. (2007). "Software quality and IS project performance improvements from software development process maturity and IS implementation strategies"; The Journal of Systems and Software; 80; pp 616-627.

Susan A. Sherer; Steven Alter; (2004a). "A general, but readily adaptable model of information system risk"; Communications of the Association for Information Systems; Vol 14; pp 1-28.

Susan A. Sherer; Steven Alter; (2004b). " Information system risks and risk factors; are they mostly about information systems?; Communications of the Association for Information Systems; Vol 14; pp 29-64.

Sauer, C. & Cuthbertson, C. (2003). The state of IT project management in the UK, Computer Weekly. 15 April.

Sauer, C., Gemino, A. & Reich, B.H. (2007). "The impact of size and volatility on IT project Performance". Communications of the ACM, 50 (10), 79–84.

Schmidt, R., Lyytinen, K., Keil, M., Cule, P., (2001). "Identifying software project risks: An international Delphi Study". Journal of Management Information Systems; Vol. 17; No.4; pp 5-36.

Seidl, D. (2007) 'The dark side of knowledge' Emergence: Complexity & Organization, 9(3), pp. 16-29.

Seyedhoseini, S. M., S. Noori & M. A. Hatefi (2009) An Integrated Methodology for Assessment and Selection of the Project Risk Response Actions. Risk Analysis, 29, 752-763.

Shakhawat, C., Husain Tahrir, Bose Neil. (2006). "Fuzzy rule-based modelling for human health risk from naturally occurring radioactive materials in produced water"; Journal of Environmental Radioactivity 89; pp.I-17.

Sherer, S.A. & Alter, S. (2004) "Information System Risk and Risk Factors: Are They Mostly About Information Systems?"; Communications of the Association for Information Systems 14, pp29- 64

Shih-Chieh Su, J., Chien-Lung Chan, Yu-Chin Liu, J., Houn-Gee Chen. (2008). "The impact of user review on software responsiveness: moderating requirements uncertainty". Information and Management 45; 203-210.

Simister, S.J. (2004). "Qualitative and quantitative risk management"; in P.W.G Morris & J.K. Pinto. (Eds); The Wiley Guide to managing projects; John Wiley; Hoboken; 2004; 30-47.

Sirkin, R.M. (2006). Statistics for the social sciences. 3rd edition. SAGE Publications.

Smith, D. 1995. The Dark Side of Excellence: Managing Strategic Failures. In J. Thompson (Ed.), Handbook of Strategic Management: 161-191. London: Butterworth-Heinemann.

Smith, D. (2005) "Business (not) as usual – crisis management, service interruption and the vulnerability of organisations"; Journal of Services Marketing, 19, 309-320.

Stevens, J.P. (1995). Applied multivariate statistics for the social sciences; 2nd edition; Hillsdale; NJ: Erlbaum Publications.

Stuart Maguire (2002). "Identifying risk during information systems development: managing the process". *Journal of Information Management & Computer Security*; Vol 10; No.3; pp 126-134.

Sumner, M. (2000). "Risk factors in enterprise-wide/ERP projects"; Journal of Information Technology, 15, 317-327.

Tan, R.R. (1996). "Success criteria and success factors for external technology transfer projects"; Project Management Journal; June; pp45-55.

Taylor, J (2003). "Managing IT Projects : Applying Project Management Strategies to Softwares, Hardwares and Integration Initiatives". AMACOM

Tbachnick, B. G. & Fidell, L.S. (2007) Using multivariate statistics, (4th ed). Boston: Allyn & Bacon.

Tiwana, A., Keil, M., (2004). "The one-minute risk assessment tool". *Communications of the ACM*; Vol 47; No.11; pp 73-77.

Tesch, D., Kloppenborg, T.J., Frolick, M.N.(2007). "IT project risk factors: The Project management professionals perspective." Journal of Computer Information Systems; 47, pp61-69.

Tom Addison. (2003). "E-commerce project development risks : evidence from a Delphi survey". International Journal of Information Management; Vol. 23; pp 23-40.

Thomas, A. V., Kalidindi, S. N. & Ganesh, L. S. (2006)." Modelling and assessment of critical risks in BOT road projects; Construction Management and Economics; 24(4); 407–24.

Tuman, J. (1993). "Project management decision making and risk management in a changing corporate environment". Project Management Institute 24th Annual Seminar; 17-19th October; Vancouver; pp. 733-739.

Turner, B. A. 1994. "The causes of disaster: Sloppy management"; British Journal of Management, 5: 215-219.

Vavpotic, D., Bajec, M. (2009). "An approach of concurrent evaluation of technical and social aspects of software development methodologies"; Information and Software Technology; 51; pp528-545.

Wastell, D., Newman, M. (1993). "The behavioral dynamics of information systems development: a stress perspective"; Accounting, Management & Information Technology 3 2 (1993), pp. 121–148.

Willcocks, L., Griffiths, M. (1997). "Information system at work: people, politics and technology"; McGraw Hill.

Verner, J.M., W.M.Evanco, N.Cerpa (2007). "State of the practice : An exploratory analysis of schedule estimation and software project success prediction". *Journal of Information & Software Technology*; Vol. 49; pp 181-193.

Wallace, L., Keil, M., & Rai, A. (2004): How Software Project Risk Affects Project Performance: an investigation of the dimensions of risk and exploratory model. Decision Sciences, 35 (2), 289-321.

Wallace, L., Mark Keil, Arun Rai (2004). "Understanding software project risk : a cluster analysis". *Journal of Information & Management;* Vol.42; pp 115-125.

Wallace, L. & Mark Keil (2004). "Software project risks and their effect on outcomes". *Communications of the ACM*; Vol 47; No.4.

Walsh, K., Schneider, H. (2002). "The role of motivation and risk behavior in software development success." Information research 7(3).

Wang, S.Q, Dulaimi, M.F, Aguria, M.Y. (2004), "Risk Management framework for construction projects in developing countries". *Construction Management & Economics.*

Wetland, J. (2006). Project Management Life cycle. Kogan Page Limited. London (E-Books).

Willcocks, L. (2000). "Risk mitigation in IT outsourcing strategy revisited: Longitudinal case research at LISA". *Journal of Strategic Information Systems*. Vol.8, pp 285-314.

Willcocks, L., Graeser, J. (2001). "Delivering IT & E-business value". *Computer Weekly Series*; Butterworth Heinemann; Oxford.

Williams, T., (1995). "A classified bibliography of recent research relating to risk management". European Journal of Operational Research; Vol 85; No.1; pp 18-38.

Yetton, P., Martin, A., Sharma, R., & Johnston, K. (2000) "A model of information systems development project performance"; Information Systems Journal, 10, 263-289.

Ying-Ming Wang, Taha M.S. Elhag. (2007). "Fuzzy group decision making approach for bridge risk assessment"; Computers & Industrial Engineering; Vol 53; Issue 1; Aug 2007; pp 137-148.

Ying-Ming Wang, Taha M.S. Elhag. (2008). "An adoptive neuro-fuzzy inference system for bridge risk assessment"; Expert Systems with Applications; Vol 34; Issue 4; May 2008; pp 3099-3106.

Yu, J; Xu, B., & Hu, H. (2007). Towards Capability Maturity in Software Review. Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International , 1, pp629 – 630.

Zadeh L.A. (1965). "Fuzzy sets". Information Control; 8; pp338-353.

Zadeh, L.A. (1994). "Fuzzy logic, neural networks, and soft computing"; Communications of the ACM; 37(3); pp77-84.

Zeng, J., Smith. NJ., (2007). "Application of a fuzzy based decision making methodology to construction project risk assessment"; International Journal of Project Mangement; 25; pp589-600.

Zhang, G.P. Keil, M. Rai, A. Mann, J. (2003). "Predicting IT project escalation : A neural network approach". *European Journal of Operational Research*. Vol. 146, pp 115-129.

Zhi, H. (1994). "Risk management for overseas construction projects". International Journal of Project Management; Vol. 13; No.3; pp.231-237.

Zhiwei Xu, Taghi M., Khoshgoftaar, Edward B. Allen. (2003). "Application of fuzzy expert systems in assessing operational risk of software", Information and Software Technology 45; pp373-388.

Zhiwei Xu. (2001). "Fuzzy logic techniques for software reliability engineering"; PhD Dissertation, Florida Atlantic University, Boca Raton, Florida, USA, May 2001

Association for Project Management (2004), *(PRAM) Project Risk Analysis and Management Guide*, Association for Project Management, High Wycombe, .

Chapman, C., Ward, S. (2003), *Project Risk Management – Processes, Techniques and Insights*, Wiley, Chichester,

Cooper, D., Grey, S., Raymond, G., Walker, P. (2005), *Project Risk Management Guidelines: Managing Risk in Large Projects and Complex Procurement*, Wiley, Chichester,

HM Treasury (2004), "The orange book", available at: www.hm-treasury.gov.uk/media/3/5/FE66035B-BCDC-D4B3-11057A7707D2521F.pdf,

Hällgren, M. (2007), "Beyond the point of no return: on the management of deviations", *International Journal of Project Management*, Vol. 25 No.8, pp.773-80.

Hillson, D. (2002), "Extending the risk process to manage opportunities", *International Journal of Project Management*, Vol. 20 No.3, pp.235-40.

Jaafari, A. (2001), "Management of risks, uncertainties and opportunities on projects: time for a fundamental shift", *International Journal of Project Management*, Vol. 19 No.2, pp.89-101.

Kristensen, V., Aven, T., Ford, D. (2006), "A new perspective on Renn and Klinke's approach to risk evaluation and management", *Reliability Engineering & System Safety*, Vol. 91 No.4, pp.421-32.

Office of Government Commerce (2002), *Managing Successful Projects with PRINCE2*, TSO, London,

Olsson, R. (2007), "In search of opportunity management: is the risk management process enough?", *International Journal of Project Management*, Vol. 25 No.8, pp.745-52.

Payne, J.H., Turner, J.R. (1999), "Company-wide project management: the planning and control of programmes of projects of different types", *International Journal of Project Management*, Vol. 17 No.1, pp.55-9.

Pellegrinelli, S., Partington, D., Hemingway, C., Mohdzain, Z., Shah, M. (2007), "The importance of context in programme management: an empirical review of programme practices", *International Journal of Project Management*, Vol. 25 No.1, pp.41-55.

Perminova, O., Gustafsson, M., Wikström, K. (2008), "Defining uncertainty in projects a new perspective", *International Journal of Project Management*, Vol. 26 No.1, pp.73-9.

Ward, S., Chapman, C. (2003), "Transforming project risk management into project uncertainty management", *International Journal of Project Management*, Vol. 21 No.2, pp.97-105.

Zhang, H. (2007), "A redefinition of the project risk process: using vulnerability to open up the event-consequence link", *International Journal of Project Management*, Vol. 25 No.7, pp.694-701.

Appendix A

Table 1.1. Software project failures

YEAR	COMPANY	OUTCOME (COSTS IN US $)
2005	Hudson Bay Co. [Canada]	Problems with inventory system contribute to $33.3 million* loss.
2004–05	UK Inland Revenue	Software errors contribute to $3.45 billion* tax-credit overpayment.
2004	Avis Europe PLC [UK]	Enterprise resource planning (ERP) system canceled after $54.5 million† is spent.
2004	Ford Motor Co.	Purchasing system abandoned after deployment costing approximately $400 million.
2004	J Sainsbury PLC [UK]	Supply-chain management system abandoned after deployment costing $527 million.†
2004	Hewlett-Packard Co.	Problems with ERP system contribute to $160 million loss.
2003–04	AT&T Wireless	Customer relations management (CRM) upgrade problems lead to revenue loss of $100 million.
2002	McDonald's Corp.	The Innovate information-purchasing system canceled after $170 million is spent.
2002	Sydney Water Corp. [Australia]	Billing system canceled after $33.2 million† is spent.
2002	CIGNA Corp.	Problems with CRM system contribute to $445 million loss.
2001	Nike Inc.	Problems with supply-chain management system contribute to $100 million loss.
2001	Kmart Corp.	Supply-chain management system canceled after $130 million is spent.
2000	Washington, D.C.	City payroll system abandoned after deployment costing $25 million.
1999	United Way	Administrative processing system canceled after $12 million is spent.
1999	State of Mississippi	Tax system canceled after $11.2 million is spent; state receives $185 million damages.
1999	Hershey Foods Corp.	Problems with ERP system contribute to $151 million loss.
1998	Snap-on Inc.	Problems with order-entry system contribute to revenue loss of $50 million.
1997	U.S. Internal Revenue Service	Tax modernization effort canceled after $4 billion is spent.
1997	State of Washington	Department of Motor Vehicle (DMV) system canceled after $40 million is spent.
1997	Oxford Health Plans Inc.	Billing and claims system problems contribute to quarterly loss; stock plummets, leading to $3.4 billion loss in corporate value.
1996	Arianespace [France]	Software specification and design errors cause $350 million Ariane 5 rocket to explode.
1996	FoxMeyer Drug Co.	$40 million ERP system abandoned after deployment, forcing company into bankruptcy.
1995	Toronto Stock Exchange [Canada]	Electronic trading system canceled after $25.5 million** is spent.
1994	U.S. Federal Aviation Administration	Advanced Automation System canceled after $2.6 billion is spent.
1994	State of California	DMV system canceled after $44 million is spent.
1994	Chemical Bank	Software error causes a total of $15 million to be deducted from 100 000 customer accounts.
1993	London Stock Exchange [UK]	Taurus stock settlement system canceled after $600 million** is spent.
1993	Allstate Insurance Co.	Office automation system abandoned after deployment, costing $130 million.
1993	London Ambulance Service [UK]	Dispatch system canceled in 1990 at $11.25 million**; second attempt abandoned after deployment, costing $15 million.**
1993	Greyhound Lines Inc.	Bus reservation system crashes repeatedly upon introduction, contributing to revenue loss of $61 million.
1992	Budget Rent-A-Car, Hilton Hotels, Marriott International, and AMR [American Airlines]	Travel reservation system canceled after $165 million is spent.

Source : Charette (2005). "Why software fails"; IEEE Spectrum

Appendix B – List of risk factors

STAGE 1 : FEASIBILITY STUDY (F)	Reference
Risk Factors	
1) Inproper justification of cost benefit analysis and evaluation criteria from feasibility study	F1
2) Too narrow focus on the technical IT issues	F2
3) Overlooked the management and business impact issues	F3
4) Wrong justification of investment alternatives and opportunity cost	F4
5) Inappropriate technology chosen from the feasibility study	F5

STAGE 2 : PROJECT PLANNING (P)	Reference
Risk Factors	
1) Unclear project scope + objectives	P1
2) Undefined project success criteria	P2
3) Lack of quality control procedure and mechanism	P3
4) Project milestones for stages not well establish	P4
5) Improper change management planning	P5
6) Inaccurate estimate of resources	P6
7) Unrealistic project schedule	P7
8) Inadequate detail breakdown structure	P8
9) Critical and non-critical activities of project not identified	P9
10) Project management and development team not properly set up	P10
11) Unclear line of decision making authority throughout the project	P11
12) Lack of contingency plan/back up	P12
13) System conversion method not well planned	P13
14) Improper planning of timeframe for project reviews and updating	P14

STAGE 3 : REQUIREMENTS (R)	Reference
Risk Factors	
1) Unclear and inadequate identification of systems requirements	R1
2) Incorrect systems requirements	R2
3) Misinterpretations of the systems requirements	R3
4) Conflicting system requirements	R4
5) Gold platting or unnecessary functions and requirements	R5
6) Inadequate validation of the requirements	R6
7) Lack of users involvement in requirement stage	R7

STAGE 4 : DEVELOPMENT (D)	Reference
Risk Factors	
1) Inproper handover from the requirement team	D1
2) Inappropriate development methodology used	D2
3) Unsuitable working model and prototype	D3
4) Programming language and CASE tool selected not adequate	D4
5) High level of technical complexities	D5
6) Project involves the use of new technology	D6
7) Difficulty in defining the input and output of system	D7
8) Immature technology	D8
9) Technological advancements and changes	D9
10) Failures and inconsistencies of unit/modules test results	D10
11) Failure of user acceptance test	D11
12) Time consuming for testing	D12
13) Resources shifted from project due to organisational priorities	D13
14) Changes in management of organisation during development	D14
15) Lack of users involvement and commitment	D15
16) Team members lack specialized skills required for the project	D16
17) Ineffective communication within development team members	D17

18) Ineffective communication between users and development team members	D18
19) Inadequately trained development team members	D19
20) Team members not familiar with the tasks/processes being *developed*	*D20*
21) Inexperienced team members	D21
22) Lack of commitment to project among development team members	D22
23) Ineffective and inexperienced project manager	D23
24) Frequent staff turnover within project team	D24
25) Conflicts between users and development team members	D25
26) Conflict among users	D26
27) Conflicts within development team members	D27
28) Excessive schedule pressure and overworked	D28
29) Lack of control and coordination within the project	D29
30) Overreliance on subcontractor or vendors/suppliers	D30
31) Redundancies and overlapping of activities/processes	D31
32) Lack of regular reviews against goals	D32
33) Large project size	D33
34) Tracking of problems within the processes/activities	D34
35) Improper sequential of processes/activities	D35

STAGE 5 : IMPLEMENTATION (IM)	Reference
Risk Factors	
1) Unsuitable conversion/installation method	IM1
2) Loss of data during conversion/installation	IM2
3) System failure during conversion/installation	IM3
4) Loss of performance during installation	IM4
5) Improper implementation sequence modules/activities	IM5
6) Disruption to existing operation/processes	IM6
7) Difficulty in configuration of system and computer environment/platform	IM7
8) Time constraints in implementation	IM8
9) Large number of interfaces to other system required	IM9
10) Users adaptability to new system	IM10
11) Users lack understanding of system capabilities and limitations	IM11
12) Failure to manage end-user expectations	IM12
13) User resistance to change	IM13
14) Feedback from users not properly analyzed	IM14
15) Time constraints of training	IM15
16) Outlining training schedule	IM16
17) Lack of knowledge and experience of system administrator/configuration manager	IM17
18) Lack of knowledge and experience of implementation team	IM18
19) Ineffective communication between users and implementation team members	IM19
20) Ineffective communication within implementation team members	IM20
21) Changes in management of organisation during implementation	IM21
22) Resources shifted from project due to organisational priorities	IM22
23) Projects affects large number of user departments/units	IM23
24) Inadequate documentation for implementation	IM24

STAGE 6 : OPERATION & MAINTENANCE (OP)	Reference
Risk Factors	
1) Lack of organisation's commitment throughout project life	OP1
2) Systems not performing accurately and effectively	OP2
3) System failure and breakdown	OP3
4) Inconsistencies of output produced	OP4
5) Inadequate user documentation	OP5
6) Poor maintenance schedule	OP6
7) Poor maintenance procedure	OP7
8) Lack of technical support	OP8
9) Threat of hackers	OP9
10) Viruses/bugs	OP10
11) Unauthorised user/sabotaj/abuse	OP11
12) Inadequate safety/security features	OP12
13) Changes in market condition and organisation priorities	OP13
14) Systems and programming languages become obsolete	OP14
15) Actions taken by competitors	OP15
16) Software not flexible in supporting new requirements and changing user needs	OP16
17) Cost of training	OP17
18) Lack of continuous IT investment to sustain competitiveness	OP18
19) Price fluctuations of hardware and software	OP19

Appendix C – Literature list of risk factors

No	Year	Title	Authors	Journals/Books/ Proceedings	References	Risk Factors
1	2007	Software development risk and project performance measurement: Evidence in Korea	Kwan Sik Na; James T.Simpson; Xiaotong Li; Tusher Singh; Ki Yoon Kim	Journal of Systems and Software (80), pp596-605	Na et al (2007)	F2-4; P1,5,6,12; R1 D2,6,15,16, IM8,10,12; OP9-11
2	2007	Evaluating software project portfolio risks	Helio R.Costa; Marcio de O.Barros; Guilherme H.Travassos	Journal of Systems and Software (80), pp16-31	Costa et al (2007)	P2,4,6,7,10; R1-5; D5-8; D13-30; D33; IM8,10,12; IM19,20,22; OP1;
3	2007	An empirical analysis of risk components and performance on software projects	Wen Ming Han; Sun Jen Huang	Journal of Systems and Software (80), pp42-50	Han and Huang (2007)	R1-3; D5-8; D14-19; D23-29; P4,6,10; IM19-22
4	2007	Managing risk in software development projects : case study	Prasanta Kumar Dey; Jason Kinch; Stephen Ogunlana.	Industrial Management and Data Systems; Vol 107; No.2; pp284-303	Dey et al (2007)	F2-4; P1,5,6,12; R2,3 D5,6,30;
5	2007	A review of techniques for risk management in projects	Ammar Ahmed; Berman Kayis; Satapom Amornsawadwatana	Benchmarking: An international Journal; Vol 14; No.1; pp 22-36	Ahmed et al (2007)	P1,5,6,7; R4; D2,6,15,16, 23,26; IM11;
6	2005	Potential risks to e-commerce developments using exploratory factor analysis	F.K.T.Wat; E.W.T.Ngai; T.C.E.Cheng	International Journal Services Technology and Management; Vol.6; No.1	Wat et al (2005)	P1,5,6,12; R3; D5,6,30; OP9-11
7	2005	The enigma of evaluation: benefits, costs and risks of IT in Australian SMEs	P.E.D.Love; Zahir Irani; Craig Standing; Chad Lin; Janice M.Burn.	Information and Management; 42; pp947-964	Love et al (2005)	P2; R1; D5,16; IM12; OP6,9-12
8	2004	Understanding software project risk: cluster analysis	Linda Wallace; Mark Keil; Arun Rai	Information and Management; 42; pp115-125	Wallace et al (2004a)	P2,4,6,7,10; R1-5; D5-8; D13-30; D33; IM8,10,12; IM19,20,22; OP1;
9	2004	Project risk management: lessons learned from software development environment	Y.H.Kwak; J.Stoddard	Technovation; 24; pp915-920	Kwak and Stoddard (2004)	F1-3; P2,4,8; R1,2,3; D18,23;
10	2004	Software project risks and their effect on outcomes	Linda Wallace; Mark Keil	Communications of the ACM; April 2004; Vol.47; No.4.	Wallace and Keil (2004)	P2,4,6,7,10; R1-5; D5-8; D13-30; D33; IM8,10,12; IM19,20,22; OP15-OP19;

11	2004	Management of risks in IT projects	David Baccarini; Geoff Salm; PED Love	Industrial Management and Data Systems; Vol.104; No.4; pp286-295	Baccarini et al (2004)	P5,10; R1,3,6; D23-27; IM24; OP5;
12	2004	Supporting risks in software project management	Marcio de Oliviera Barros; Claudia Maria Lima Werner; Guilherme Horta Travassos;	Journal of Systems and Software; 70; pp21-35	Barros et al (2004)	P1,2,4,6,7,8,12; R8; D5,15,29;
13	2004	How software project risk affects project performance: investigation of the dimensions of risk and exploratory model	Linda Wallace; Mark Keil; Arun Rai	Decision Sciences; Vol.35; No.2	Wallace et al (2004b)	P2,4,6,7,10; R1-5; D5-8; D13-30; D33; IM8,10,12; IM19,20,22; OP1; OP15-OP19;
14	2003	Risk analysis in project of software development	Xiangnan Lu; Yali Ge;	IEEE	Lu and Ge (2003)	P2,4,8; R1,2,3; D18,23;
15	2003	Why software projects escalate: Importance of project management construct	Mark Keil; Arun Rai; Joan Ellen Cheney Mann; G.Peter Zhang;	IEEE	Keil et al (2003)	P1,2,4,6,7,8,12; R8; D5,15,29;
16	2003	E-commerce project development risks: evidence form Delphi survey	Tom Addison	International Journal of Information Management; 23; pp25-40	Addison (2003)	F2-4; R4; IM11; OP12;
17	2003	Predicting IT project escalation: neural network approach	G.Peter Zhang; Mark Keil; Arun Rai; Joan Mann;	European Journal of Operational Research; 146; pp115-129	Zhang et al (2003)	P1,2,4,6,7,8,12; R8; D5,15,29;
18	2002	Managing risks in IT project: an options perspective	Ram.L.Kumar	Information and Management; 40; pp63-74	Kumar (2002)	P6,7; R1,2; OP17;
19	2002	Reducing user related risks during and prior to system development	James J.Jiang; Gary Klein; Hong Gee Chen; Laura Lin;	International Journal of Project Management; 20; pp507-515	Jiang et al (2002)	R8; D11-15; IM9,10,12;
20	2002	Identifying risks during IS development: managing the process	Stuart Maguire	Information Management and Computer Security; 10(3); pp126-134	Maguire (2002)	F1; OP9-12;
21	2002	Reconciling user and project manager perceptions of IT project risk: Delphi study	Mark Keil; Amrit Tiwana; Ashley Bush	Information Systems Journal; 12; pp103-119	Keil et al (2002)	P1,5,6,7; R4; D2,6,15,16, 23,26; IM11;
22	2002	Case study: factors for early prediction of software development success.	J.Drew Procaccino; June M.Verner; Scott P.Overmyer; Marvin E.Parter;	Information and Software Technology; 44; pp53-62	Procaccino et al (2002)	P6; R3,4; D6,15,16,19,25,26; IM11;
23	2000	Components of software development risk: how to address them; Project Manager survey	Janne Ropponen; Kalle Lyytinen	IEEE Transactions on Software Engineering; Vol.26; No.2; Feb	Ropponen and Lyytinen (2000)	P7; R3,6; D5;
24	2000	Software development risks to project effectiveness	James Jiang; Gary Klein;	Journal of Systems and Software; 52; pp3-10;	Jiang and Klein (2000)	P6; D15,16,25,33
25	1999	Risk to different aspects of system success	James Jiang; Gary Klein;	Information and Management; 36; pp263-272	Jiang and Klein (1999)	P6; D5,6,15,16, 33;

26	1998	A framework for identifying software project risks	Mark Keil; Paul E.Cule; Kalle Lyytinen; Roy C.Schmidt;	Communications of the ACM; Nov 1998; Vol.41; No.11;	Keil et al (1998)	P6; R3,4; D6,15,16,19,25,26; IM11;
27	1997	Implementing risk management on software intensive projects	Edmund H.Conrow; Patricia S.Shishido;	IEE Software; May/June 1997	Conrow and Shishido (1997)	P6,7; R1; D5,15,18;
28	1996	Risks in the use of IT within organizations	G.Dhillon; J.Backhouse;	International Journal of Information Management; Vol.16; No.1; pp65-74	Dhillon and Backhouse (1996)	P1,2,4,6,7,8,12; R8; D5,15,29; OP9-12;
29	1996	Risk analysis techniques and their application to software development	Joanna C.Bennett; George A.Bohoris; Elaine M.Aspinwall; Richard C.Hall;	European Journal of Operational Research; 95; pp467-475;	Bennett et al (1996)	P6; R3,4; D6,15,16,19,25,26; IM11;
30	1996	A business approach to effective IT risk analysis and management	Sharon Halliday; Karen Badenhorst; Rossouw von Solms;	Information Management and Computer Security; 4(1); pp19-31;	Halliday (1996)	P6,7; R2,3,6; D3,10,16: OP12;
31	1994	Predicting risk of failure in large scale IT projects	Leslie Willcocks; Catherine Griffiths	Technological Forecasting and social change; 47; pp205-228	Willcocks and Griffiths (1994)	P1,5,6,7; R4; D2,6,15,16,23,26; IM11;
32	1991	Software risk management : principles and practices	Barry W.Boehm	IEEE Software; Jan 1991	Boehm (1991)	P6,7; R2,3,6; D3,10,16:

Appendix D

Table 5.1. Mean average rating for Likelihood of occurrence for all 104 risk factors

					Ranking							
Stages	Risk factors	Ref	Mean	Standard Deviation	Board	Project Manager	Developer	IT staff	Coefficient of variation	Severity index	Kendall ranking	Overall Ranking
Feasibility study	Inproper justification of cost benefit analysis and evaluation criteria from feasibility study	F1	3.60	0.94	5	23	29	33	26.11	72.04	71.97	21
	Too narrow focus on technical IT issues	F2	3.34	0.871	12	43	26	61	26.07	66.85	67.22	32
	Overlooked the management and business impact issues	F3	3.47	1.06	31	15	35	45	30.54	69.44	67.54	24
	Wrong justification of investment alternatives and opportunity cost	F4	2.28	0.835	68	66	69	64	36.6	45.62	45.03	67
	Inappropriate technology chosen from the feasibility study	F5	2.44	0.862	60	60	62	56	35.32	48.83	50.39	61
Project planning	Unclear project scope + objectives	P1	4.17	1.07	6	6	5	4	25.65	83.4	84.20	4
	Undefined project success criteria	P2	3.69	0.829	20	19	18	17	22.46	73.7	77.24	19
	Lack of quality control procedure and mechanism	P3	2.72	0.964	45	50	53	40	35.44	54.32	54.30	51
	Project milestones for stages not well establish	P4	2.41	0.826	57	63	61	48	34.27	48.27	44.18	62
	Improper change management planning	P5	3.56	0.976	24	22	25	26	27.41	71.17	70.48	22
	Inaccurate estimate of resources	P6	4.44	0.854	1	3	1	2	19.23	88.7	91.84	1
	Unrealistic project schedule	P7	4.41	0.784	2	2	2	3	17.77	88.21	92.18	2
	Inadequate detail work breakdown structure (WBS)	P8	2.56	0.986	51	56	56	44	38.51	51.11	48.95	55
	Critical and non-critical activities of project not identified	P9	3.73	1.019	11	12	22	55	27.31	74.69	75.48	16
	Project management and development team not properly set up	P10	1.81	1.107	86	89	89	85	61.16	36.17	28.89	89
	Unclear line of decision making authority throughout the project	P11	3.99	0.897	21	1	23	22	22.48	79.88	82.15	9
	Lack of contingency plan/back up	P12	3.66	0.615	13	21	21	21	16.80	73.27	73.79	20
	System conversion method not well planned	P13	3.30	0.995	25	28	34	72	30.15	66.05	63.67	34
	Improper planning of timeframe for project reviews and updating	P14	3.19	0.830	36	33	40	47	26.01	63.89	64.35	41

Requirement	Unclear and inadequate identification of systems requirements	R1	4.08	0.767	10	9	8	7	18.79	81.6	84.02	7
	Incorrect systems requirements	R2	2.15	1.32	73	71	74	70	61.39	42.96	37.53	73
	Misinterpretations of the systems requirements	R3	3.32	0.852	23	26	36	66	25.66	66.36	68.85	33
	Conflicting system requirements	R4	2.44	0.986	64	57	64	59	40.41	48.89	49.92	60
	Gold platting or unnecessary functions and requirements	R5	1.90	1.040	87	80	86	93	54.73	37.96	34.05	85
	Inadequate validation of the requirements	R6	3.82	0.974	42	11	11	10	25.49	76.36	77.60	12
	Lack of users involvement in requirement stage	R7	3.41	1.658	50	24	24	37	48.6	68.21	64.75	28
Development	Inproper handover from the requirement team	D1	3.14	1.056	30	44	47	27	33.6	62.72	63.72	43
	Inappropriate development methodology used	D2	4.16	0.854	18	4	3	9	20.52	83.21	86.57	5
	Unsuitable working model and prototype	D3	2.53	0.830	52	58	57	46	32.8	50.62	47.52	56
	Programming language and CASE tool selected not adequate	D4	2.51	1.139	53	59	58	51	45.37	50.25	46.79	57
	High level of technical complexities	D5	3.79	0.944	48	13	9	13	24.9	75.8	78.36	13
	Project involves the use of new technology	D6	2.34	1.036	66	62	66	71	44.27	46.79	43.81	65
	Difficulty in defining the input and output of system	D7	3.09	1.175	28	41	45	75	38.02	61.73	61.54	45
	Immature technology	D8	2.22	0.887	70	67	49	73	39.95	44.44	40.61	70
	Technological advancements and changes	D9	2.06	0.819	76	77	76	78	39.75	41.23	35.25	76
	Failures and inconsistencies of unit/modules test results	D10	2.25	0.975	67	72	68	62	43.33	45.06	38.69	68
	Failure of user acceptance test	D11	3.21	0.937	38	36	38	35	29.19	64.2	62.27	40
	Time consuming for testing	D12	3.69	0.849	15	20	20	18	23.01	73.89	77.11	18
	Resources shifted from project due to organisational priorities	D13	3.35	1.047	4	38	43	29	31.25	67.04	64.45	31
	Changes in management of organisation during development	D14	1.97	1.007	77	87	78	67	51.11	39.38	31.04	79
	Lack of users involvement and commitment	D15	3.92	1.139	9	14	14	8	29.05	78.46	77.78	11
	Team members lack specialized skills required for the project	D16	2.48	1.176	58	54	60	68	47.41	49.63	48.57	59
	Ineffective communication within development team members	D17	4.3	0.9	3	5	4	1	20.93	85.93	88.01	3
	Ineffective communication between users and development team members	D18	3.37	0.982	29	29	32	30	29.13	67.35	65.24	30
	Inadequately trained development team members	D19	1.96	1.052	78	83	79	79	53.67	39.14	30.99	81

	Team members not familiar with the tasks/processes being developed	D20	1.92	0.725	83	84	84	83	37.76	38.4	29.88	84
	Inexperienced team members	D21	2.15	1.179	74	69	72	80	54.83	42.96	40.38	72
	Lack of commitment to project among development team members	D22	2.90	1.036	39	48	50	34	36.72	57.9	55.62	49
	Ineffective and inexperienced project manager	D23	4.02	0.721	14	8	10	12	17.93	80.37	82.59	8
	Frequent staff turnover within project team	D24	3.38	0.973	34	46	13	19	28.78	67.53	67.40	28
	Conflicts between users and development team members	D25	2.13	0.925	72	76	75	57	43.42	42.59	35.99	74
	Conflict among users	D26	1.75	1.048	91	95	93	84	59.88	35	28.60	91
	Conflicts within development team members	D27	1.60	0.973	99	99	100	102	60.81	31.91	23.19	100
	Excessive schedule pressure and overworked	D28	3.98	0.924	40	7	6	6	23.21	79.63	80.62	10
	Lack of control and coordination within the project	D29	3.79	0.996	17	16	17	15	26.27	75.74	75.03	14
	Overreliance on subcontractor or vendors/suppliers	D30	1.61	0.971	100	97	99	99	60.31	32.28	26.40	99
	Redundancies and overlapping of activities/processes	D31	2.05	0.925	79	75	77	74	45.12	41.05	36.39	77
	Lack of regular reviews against goals	D32	4.09	0.808	8	10	7	5	19.70	81.73	83.43	6
	Large project size	D33	2.25	0.964	69	70	70	52	42.84	44.94	39.33	69
	Tracking of problems within the processes/activities	D34	1.52	0.500	101	101	101	100	32.89	30.43	22.15	101
	Improper sequential of processes/activities	D35	1.98	0.862	80	79	82	77	43.53	39.63	34.30	78
Implementation	Unsuitable conversion/installation method	IM1	1.94	0.744	81	81	80	81	38.35	38.89	31.91	82
	Loss of data during conversion/installation	IM2	2.30	0.940	65	68	67	53	40.86	45.93	41.90	66
	System failure during conversion/installation	IM3	1.78	0.819	88	92	90	89	46.01	35.68	26.42	90
	Loss of performance during installation	IM4	2.10	0.841	75	74	73	76	40.04	42.1	37.27	75
	Improper implementation sequence modules/activities	IM5	1.67	0.851	92	100	97	94	50.95	33.33	24.13	98
	Disruption to existing operation/processes	IM6	1.69	0.658	93	98	96	86	38.93	33.7	24.26	96
	Difficulty in configuration of system and computer environment/platform	IM7	1.94	0.744	82	82	81	82	38.35	38.89	31.91	83
	Time constraints in implementation	IM8	3.00	0.946	56	42	44	41	31.53	60.06	57.43	47

	Large number of interfaces to other system required	IM9	3.37	1.016	35	47	12	14	30.14	67.41	67.56	29
	Users adaptability to new system	IM10	1.88	1.015	84	88	85	90	53.98	37.53	27.59	88
	Users lack understanding of system capabilities and limitations	IM11	2.37	0.998	61	65	65	60	42.10	47.41	42.03	64
	Failure to manage end-user expectations	IM12	3.19	1.553	54	32	31	63	48.68	63.77	57.19	42
	User resistance to change	IM13	2.70	1.039	43	52	52	42	38.48	53.95	48.68	52
	Feedback from users not properly analyzed	IM14	3.22	0.942	33	34	37	39	29.25	64.32	61.20	37
	Time constraints of training	IM15	3.79	0.841	16	17	15	16	22.18	75.74	78.29	15
	Outlining training schedule	IM16	3.13	1.312	63	39	41	23	41.91	62.53	57.72	44
	Lack of knowledge and experience of system administrator/configuration manager	IM17	2.39	0.823	62	64	63	54	34.43	47.78	42.16	63
	Lack of knowledge and experience of implementation team	IM18	2.51	0.962	55	61	59	49	38.32	50.12	45.36	58
	Ineffective communication between users and implementation team members	IM19	3.46	0.958	26	27	28	28	27.68	69.26	67.97	25
	Ineffective communication within implementation team members	IM20	3.07	0.918	44	40	42	50	29.90	61.48	57.46	46
	Changes in management of organisation during implementation	IM21	2.81	1.209	41	51	51	31	43.02	56.17	51.44	50
	Resources shifted from project due to organisational priorities	IM22	3.48	0.871	27	25	27	24	25.02	69.63	68.63	23
	Projects affects large number of user departments/units	IM23	3.71	1.015	19	18	19	20	27.35	74.2	76.04	17
	Inadequate documentation	IM24	3.25	1.044	46	49	16	11	32.12	64.94	65.55	35
Operation Maintenance	Lack of organisation's commitment throughout project life	OP1	3.21	1.102	22	35	39	69	34.33	64.26	60.81	39
	Systems not performing accurately and effectively	OP2	2.20	0.618	71	73	71	65	28.09	43.95	39.72	71
	System failure and breakdown	OP3	1.75	0.715	94	91	92	91	40.85	34.94	28.75	93
	Inconsistencies of output produced	OP4	1.73	0.695	95	93	94	96	40.17	34.57	28.76	94
	Inadequate user documentation	OP5	2.62	0.945	49	55	55	38	36.06	52.35	47.79	54
	Poor maintenance schedule	OP6	3.22	1.429	59	30	30	58	44.37	64.38	59.29	36
	Poor maintenance procedure	OP7	2.66	0.891	47	53	54	36	33.49	53.21	51.00	53
	Lack of technical support	OP8	1.73	0.695	96	94	95	97	40.17	34.57	28.76	95
	Threat of hackers	OP9	1.45	0.730	102	103	103	103	50.34	29.01	20.98	104

	Viruses/bugs	OP10	1.45	0.730	103	104	104	104	50.34	29.01	20.98	103
	Unauthorised user/sabotaj/abuse	OP11	1.47	0.500	104	102	102	101	34.01	29.38	21.11	102
	Inadequate safety/security features	OP12	1.68	0.710	97	96	98	98	42.26	33.52	24.83	97
	Changes in market condition and organisation priorities	OP13	3.39	1.036	7	31	33	43	30.56	67.72	64.85	27
	Systems and programming languages become obsolete	OP14	1.85	0.807	89	86	88	95	43.62	36.91	32.10	88
	Actions taken by competitors	OP15	1.86	0.822	90	85	87	87	44.19	37.28	32.09	87
	Software not flexible in supporting new requirements and changing user needs	OP16	2.99	1.026	37	45	48	32	34.31	2.623	56.58	48
	Cost of training	OP17	1.96	1.013	85	78	83	92	51.68	39.26	35.30	80
	Lack of continuous IT investment to sustain competitiveness	OP18	3.22	0.627	32	37	46	25	19.47	64.32	64.65	38
	Price fluctuations of hardware and software	OP19	1.75	0.892	98	90	91	88	50.97	35	31.41	92

Appendix E **Table 5.8 : Mean average rating for Impact of risk factors on cost overrun for all 104 factors**

					Ranking							
Stages	Risk factors	Ref	Mean	Standard deviation	Board	Project Manager	Developer	IT staff	Coefficient of variation	Severity index	Kendall ranking	Overall Ranking
Feasibility study	Inproper justification of cost benefit analysis and evaluation criteria from feasibility study	F1	3.24	1.208	17	38	41	21	37.28	64.81	65.47	33
	Too narrow focus on technical IT issues	F2	2.58	0.867	46	48	48	39	33.60	51.67	51.59	48
	Overlooked the management and business impact issues	F3	3.44	0.995	27	22	20	25	28.92	68.7	69.97	20
	Wrong justification of investment alternatives and opportunity cost	F4	2.29	0.977	59	58	65	56	42.66	45.74	46.15	62
	Inappropriate technology chosen from the feasibility study	F5	2.02	0.710	71	78	84	71	35.14	40.31	39.65	77
Project planning	Unclear project scope + objectives	P1	4.33	0.572	3	2	2	2	13.21	86.6	91.72	1
	Undefined project success criteria	P2	3.67	0.809	12	16	13	15	22.04	73.46	78.71	13
	Lack of quality control procedure and mechanism	P3	2.56	0.854	44	49	51	41	33.35	51.3	51.50	49
	Project milestones for stages not well establish	P4	3.18	1.035	18	45	19	20	32.54	63.64	66.38	38
	Improper change management planning	P5	3.82	0.818	4	4	24	28	21.41	76.36	81.75	12
	Inaccurate estimate of resources	P6	4.24	0.886	5	6	1	1	20.89	84.88	89.26	3
	Unrealistic project schedule	P7	4.32	0.776	1	1	4	5	17.96	86.42	91.16	2
	Inadequate detail work breakdown structure (WBS)	P8	2.29	0.978	61	59	58	54	42.70	45.8	42.85	59
	Critical and non-critical activities of project not identified	P9	3.53	0.884	16	18	17	18	25.04	70.68	75.24	19
	Project management and development team not properly set up	P10	1.97	1.055	76	82	86	80	53.55	39.38	33.88	81
	Unclear line of decision making authority throughout the project	P11	3.04	1.131	47	41	39	53	37.20	60.86	63.14	43
	Lack of contingency plan/back up	P12	4.21	1.019	2	3	3	4	24.20	84.14	86.57	4
	System conversion method not well planned	P13	2.10	0.783	69	72	74	67	37.28	41.91	38.25	71
	Improper planning of timeframe for project reviews and updating	P14	3.27	1.516	33	30	37	24	46.36	65.49	65.76	29

Requirement	Unclear and inadequate identification of systems requirements	R1	3.99	0.867	8	7	6	6	21.72	79.81	83.38	6
	Incorrect systems requirements	R2	2.38	0.980	62	55	56	60	41.17	47.59	50.39	56
	Misinterpretations of the systems requirements	R3	3.89	0.827	9	9	9	11	21.25	77.72	81.68	9
	Conflicting system requirements	R4	1.98	0.684	85	79	81	75	34.54	39.69	38.65	80
	Gold platting or unnecessary functions and requirements	R5	1.91	0.899	86	87	90	77	47.06	38.15	37.23	88
	Inadequate validation of the requirements	R6	3.84	1.456	6	12	11	7	37.91	76.86	76.06	11
	Lack of users involvement in requirement stage	R7	3.22	0.998	30	31	32	64	30.99	64.38	65.54	34
Development	Inproper handover from the requirement team	D1	3.26	1.387	24	28	34	65	42.54	66.12	65.21	31
	Inappropriate development methodology used	D2	3.62	1.102	14	14	18	13	30.44	72.47	74.45	14
	Unsuitable working model and prototype	D3	2.29	0.983	58	64	61	52	42.92	45.74	43.21	61
	Programming language and CASE tool selected not adequate	D4	2.54	1.133	45	51	50	44	44.60	50.86	49.94	51
	High level of technical complexities	D5	3.28	1.113	19	20	43	19	33.93	65.68	70.63	26
	Project involves the use of new technology	D6	2.67	0.973	50	46	45	42	36.44	53.4	54.34	46
	Difficulty in defining the input and output of system	D7	2.31	0.968	54	60	62	49	41.90	46.11	45.40	57
	Immature technology	D8	2.06	0.825	72	74	72	81	40.04	41.3	38.21	73
	Technological advancements and changes	D9	2.10	1.211	78	71	66	83	57.66	42.04	37.59	72
	Failures and inconsistencies of unit/modules test results	D10	2.29	0.983	57	63	60	51	42.92	45.74	43.21	60
	Failure of user acceptance test	D11	4.00	1.452	21	5	5	3	36.3	79.94	77.91	4
	Time consuming for testing	D12	2.43	0.854	53	53	54	46	35.14	48.58	46.98	53
	Resources shifted from project due to organisational priorities	D13	3.20	1.061	48	21	27	55	33.15	63.95	65.52	36
	Changes in management of organisation during development	D14	2.14	1.073	70	56	68	68	50.14	42.72	38.50	69
	Lack of users involvement and commitment	D15	3.90	1.182	10	11	7	10	30.30	77.96	80.76	8
	Team members lack specialized skills required for the project	D16	2.05	0.953	82	73	70	85	46.48	40.99	37.19	75
	Ineffective communication within development team members	D17	3.12	1.458	40	37	33	48	46.73	62.47	61.78	41
	Ineffective communication between users and development team members	D18	3.24	1.210	35	32	29	37	37.34	64.75	65.19	32
	Inadequately trained development team members	D19	1.95	0.751	79	83	83	82	38.51	38.95	33.56	82

	Team members not familiar with the tasks/processes being developed	D20	1.81	0.817	89	91	91	90	45.13	36.23	29.50	90
	Inexperienced team members	D21	1.86	0.920	91	89	87	93	49.46	37.16	34.16	89
	Lack of commitment to project among development team members	D22	2.55	1.116	51	50	49	34	43.76	50.99	50.08	50
	Ineffective and inexperienced project manager	D23	3.94	0.766	7	8	8	8	19.44	78.7	81.68	7
	Frequent staff turnover within project team	D24	2.75	1.210	38	44	46	29	44.00	54.94	53.47	45
	Conflicts between users and development team members	D25	2.66	0.967	41	47	47	31	36.35	53.27	57.44	47
	Conflict among users	D26	2.52	1.114	49	52	52	38	44.20	50.49	52.88	52
	Conflicts within development team members	D27	1.48	0.706	100	100	100	100	47.70	29.69	24.36	100
	Excessive schedule pressure and overworked	D28	3.17	1.309	22	42	21	58	41.29	63.4	62.61	39
	Lack of control and coordination within the project	D29	3.62	1.296	13	15	16	16	35.80	72.35	71.35	15
	Overreliance on subcontractor or vendors/suppliers	D30	2.02	1.084	83	75	75	72	53.66	40.49	40.62	79
	Redundancies and overlapping of activities/processes	D31	1.65	0.682	92	95	96	86	40.12	33.02	28.85	95
	Lack of regular reviews against goals	D32	1.64	0.827	97	96	95	91	50.42	32.72	28.66	96
	Large project size	D33	2.03	1.086	80	76	76	73	53.49	40.56	40.73	76
	Tracking of problems within the processes/activities	D34	1.23	0.418	103	103	103	103	33.96	24.51	16.33	103
	Improper sequential of processes/activities	D35	1.71	0.812	95	93	93	87	47.48	34.2	30.03	93
Implementation	Unsuitable conversion/installation method	IM1	2.02	1.024	77	80	79	70	50.69	40.31	34.63	78
	Loss of data during conversion/installation	IM2	2.31	0.968	55	62	64	50	41.90	46.11	45.40	58
	System failure during conversion/installation	IM3	1.94	1.053	75	85	85	84	54.27	38.83	32.65	84
	Loss of performance during installation	IM4	1.93	0.904	84	86	82	88	46.83	38.52	33.65	86
	Improper implementation sequence modules/activities	IM5	1.93	0.737	87	84	80	89	38.18	38.58	33.69	85
	Disruption to existing operation/processes	IM6	2.43	0.854	56	54	55	47	35.14	48.58	46.98	54
	Difficulty in configuration of system and computer environment/platform	IM7	1.53	0.696	98	99	99	97	45.49	30.68	21.21	99
	Time constraints in implementation	IM8	3.09	1.221	42	39	36	45	39.51	61.85	63.15	42
	Large number of interfaces to other system required	IM9	3.15	1.201	32	40	42	27	38.12	62.9	66.30	40

	Users adaptability to new system	IM10	2.39	1.115	60	57	53	57	46.65	47.84	46.05	55
	Users lack understanding of system capabilities and limitations	IM11	2.15	0.944	64	70	71	61	43.90	43.02	38.70	68
	Failure to manage end-user expectations	IM12	3.58	1.569	43	13	12	12	43.82	71.6	69.22	17
	User resistance to change	IM13	3.85	1.064	11	10	10	9	27.63	76.98	79.33	10
	Feedback from users not properly analyzed	IM14	3.60	1.371	15	17	15	14	38.08	72.1	72.04	16
	Time constraints of training	IM15	3.26	1.298	28	34	38	26	39.81	65.19	65.55	30
	Outlining training schedule	IM16	2.17	0.954	73	67	63	78	43.96	43.4	39.90	67
	Lack of knowledge and experience of system administrator/configuration manager	IM17	2.27	0.977	65	61	57	59	43.03	45.49	43.05	63
	Lack of knowledge and experience of implementation team	IM18	2.19	1.007	74	65	59	79	45.98	43.77	40.92	66
	Ineffective communication between users and implementation team members	IM19	3.33	1.053	31	27	25	30	31.62	66.67	67.84	22
	Ineffective communication within implementation team members	IM20	3.31	1.114	29	29	28	32	33.65	66.17	68.57	24
	Changes in management of organisation during implementation	IM21	3.30	0.896	52	23	26	22	27.15	65.93	70.95	25
	Resources shifted from project due to organisational priorities	IM22	3.43	0.900	25	24	22	23	26.23	68.58	73.37	21
	Projects affects large number of user departments/units	IM23	3.54	1.255	20	19	14	17	35.45	70.86	75.28	18
	Inadequate documentation	IM24	2.90	0.914	36	43	44	33	31.51	58.09	59.75	44
Operation Maintenance	Lack of organisation's commitment throughout project life	OP1	3.20	1.118	37	36	35	35	34.93	64.01	64.11	35
	Systems not performing accurately and effectively	OP2	2.14	0.776	68	69	73	66	36.26	42.78	43.03	70
	System failure and breakdown	OP3	1.95	1.022	88	81	78	92	52.41	39.01	38.20	83
	Inconsistencies of output produced	OP4	3.28	1.249	23	33	40	40	38.07	65.68	66.85	27
	Inadequate user documentation	OP5	1.92	0.660	81	88	89	74	34.37	38.33	34.06	87
	Poor maintenance schedule	OP6	2.19	0.635	63	68	69	62	28.99	43.83	43.13	65
	Poor maintenance procedure	OP7	2.19	0.636	67	66	67	63	29.04	43.89	42.34	64
	Lack of technical support	OP8	1.71	0.692	90	94	94	95	40.46	34.32	30.84	94
	Threat of hackers	OP9	1.31	0.465	102	102	102	102	35.49	26.3	19.40	102
	Viruses/bugs	OP10	1.44	0.497	101	101	101	101	34.51	28.7	21.77	101

	Unauthorised user/sabotaj/abuse	OP11	2.06	1.089	86	77	77	76	52.86	41.11	38.89	74
	Inadequate safety/security features	OP12	1.57	0.720	98	98	98	98	45.85	31.42	26.29	98
	Changes in market condition and organisation priorities	OP13	3.18	0.994	39	35	30	43	31.25	63.52	66.27	37
	Systems and programming languages become obsolete	OP14	1.76	1.122	94	92	92	94	63.75	35.35	31.10	92
	Actions taken by competitors	OP15	1.58	0.723	99	97	97	99	45.75	31.54	25.99	97
	Software not flexible in supporting new requirements and changing user needs	OP16	3.32	1.083	34	25	23	36	32.62	66.48	68.67	23
	Cost of training	OP17	1.81	0.936	93	90	88	96	51.71	36.3	33.69	91
	Lack of continuous IT investment to sustain competitiveness	OP18	3.27	1.085	26	26	31	69	33.18	65.49	66.61	28
	Price fluctuations of hardware and software	OP19	1.23	0.418	104	104	104	104	33.96	24.51	16.33	104

Appendix F Geographical spread of respondents by country

	Frequency	Percent	Valid Percent	Cumulative Percent
UK	81	25.0	25.0	25.0
USA	131	40.4	40.4	65.4
Canada	18	5.6	5.6	71.0
Japan	6	1.9	1.9	72.8
Netherland	3	.9	.9	73.8
India	28	8.6	8.6	82.4
France	3	.9	.9	83.3
Australia	7	2.2	2.2	85.5
Russia	6	1.9	1.9	87.3
Finland	1	.3	.3	87.7
Romania	4	1.2	1.2	88.9
China	3	.9	.9	89.8
Ukraine	3	.9	.9	90.7
Vietnam	1	.3	.3	91.0
Lebanon	1	.3	.3	91.4
Belarus	1	.3	.3	91.7
Denmark	1	.3	.3	92.0
Germany	2	.6	.6	92.6
Switzerland	3	.9	.9	93.5
Sweden	2	.6	.6	94.1
Norway	2	.6	.6	94.8
Czech Republic	1	.3	.3	95.1
Jordan	1	.3	.3	95.4
Mexico	2	.6	.6	96.0
Italy	1	.3	.3	96.3
United Arab Emirates	1	.3	.3	96.6
Singapore	3	.9	.9	97.5
Malaysia	1	.3	.3	97.8
Hong Kong	2	.6	.6	98.5
Spain	1	.3	.3	98.8
Sri Lanka	2	.6	.6	99.4
Thailand	1	.3	.3	99.7
Phillippines	1	.3	.3	100.0
Total	324	100.0	100.0	

Appendix G - Questionnaire : Respondent Informations

SECTION A (Respondent's Information)

1. Designation :

☐ Managing Director/Board ☐ Project Manager
☐ Software developer/programmer ☐ IT staffs ☐ Others

2. Nature of business :

☐ Software development
☐ IT consultancy and management
☐ Web development
☐ Others

3. Number of software development project involved/undertaken :

☐ Less than 3 ☐ 3-6 ☐ 7-10 ☐ 11-14 ☐ 15-18 ☐ More than 18

4. Your experience in management of software development project (years) :

☐ Less than 3 ☐ 3-6 ☐ 7-10 ☐ 11-14 ☐ 15-18 ☐ More than 18

SECTION B (Risk Management criterias)

a) How many risk management expert within your organization?

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ > 7 persons

b) Are you satisfied with the risk management practice of your company?

Not satisfied → Very satisfied

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

c) How often do you carry out risk assessment for your software development project?

☐ Not al all ☐ Seldom ☐ Regular ☐ Very often

d) Do you use computerised risk assessment packages?

☐ Not al all ☐ Seldom ☐ Regular ☐ Very often

e) Based on your experience, what is the percentage (%) of cost overrun over the total overall cost of the software development project?

☐ Very low (0-10%) ☐ Low (11-20%) ☐ Moderate (21-30%)

☐ High (31-40%) ☐ Very high (>40%)

Thank you very much for your kind cooperation

Appendix H : Questionnaire - Feasibility study stage

STAGE 1 : FEASIBILITY STUDY	Likelihood Occurrence					Risk factors impact on cost overrun as % deviation from original estimate				
						Cost				
	None	Unlikely	Likely	Highly Likely	Very highly likely	Very Low 1-10% overrun	Low 11-20% overrun	Moderate 21-30% overrun	High 31-40% overrun	Very High > 40% overrun
Risk Factors	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
1) Inproper justification of cost benefit analysis and evaluation criteria from feasibility study	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2) Too narrow focus on the technical IT issues	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
3) Overlooked the management and business impact issues	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
4) Wrong justification of investment alternatives and opportunity cost	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
5) Inappropriate technology chosen from the feasibility study	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Appendix H : Questionnaire - Project Planning stage

STAGE 2 : PROJECT PLANNING	Likelihood Occurrence					Risk factors impact on cost overrun as % deviation from original estimate				
						Cost				
	None	Unlikely	Likely	Highly Likely	Very highly likely	Very Low	Low	Moderate	High	Very High
						1-10% overrun	11-20% overrun	21-30% overrun	31-40% overrun	> 40% overrun
Risk Factors	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
1) Unclear project scope + objectives	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2) Undefined project success criteria	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
3) Lack of quality control procedure and mechanism	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
4) Project milestones for stages not well establish	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
5) Improper change management planning	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
6) Inaccurate estimate of resources	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
7) Unrealistic project schedule	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
8) Inadequate detail breakdown structure	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
9) Critical and non-critical activities of project not identified	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
10) Project management and development team not properly set up	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
11) Unclear line of decision making authority throughout the project	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
12) Lack of contingency plan/back up	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
13) System conversion method not well planned	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
14) Improper planning of timeframe for project reviews and updating	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Appendix H : Questionnaire - Requirement stage

STAGE 3 : REQUIREMENTS	Likelihood Occurrence					Risk factors impact on cost overrun as % deviation from original estimate				
						Cost				
	None	Unlikely	Likely	Highly Likely	Very highly likely	Very Low	Low	Moderate	High	Very High
						1-10% overrun	11-20% overrun	21-30% overrun	31-40% overrun	> 40% overrun
Risk Factors	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
1) Unclear and inadequate identification of systems requirements	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2) Incorrect systems requirements	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
3) Misinterpretations of the systems requirements	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
4) Conflicting system requirements	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
5) Gold plating or unnecessary functions and requirements	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
6) Inadequate validation of the requirements	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
7) Lack of users involvement in requirement stage	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Appendix H : Questionnaire - Development stage

STAGE 4 : DEVELOPMENT	Likelihood Occurrence					Risk factors impact on cost overrun as % deviation from original estimate: Cost				
	None	Unlikely	Likely	Highly Likely	Very highly likely	Very Low 1-10% overrun	Low 11-20% overrun	Moderate 21-30% overrun	High 31-40% overrun	Very High > 40% overrun
Risk Factors	1	2	3	4	5	1	2	3	4	5
1) Inproper handover from the requirement team	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2) Inappropriate development methodology used	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
3) Unsuitable working model and prototype	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
4) Programming language and CASE tool selected not adequate	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
5) High level of technical complexities	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
6) Project involves the use of new technology	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
7) Difficulty in defining the input and output of system	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
8) Immature technology	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
9) Technological advancements and changes	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
10) Failures and inconsistencies of unit/modules test results	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
11) Failure of user acceptance test	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
12) Time consuming for testing	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
13) Resources shifted from project due to organisational priorities	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
14) Changes in management of organisation during development	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
15) Lack of users involvement and commitment	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
16) Team members lack specialized skills required for the project	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
17) Ineffective communication within development team members	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
18) Ineffective communication between users and development team members	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
19) Inadequately trained development team members	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
20) Team members not familiar with the tasks/processes being developed	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
21) Inexperienced team members	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
22) Lack of commitment to project among development team members	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Appendix H : Questionnaire - Development stage (continued)

STAGE 4 : DEVELOPMENT (continued)	Likelihood Occurrence					Risk factors impact on cost overrun as % deviation from original estimate				
						Cost				
	None	Unlikely	Likely	Highly Likely	Very highly likely	Very Low	Low	Moderate	High	Very High
						1-10% overrun	11-20% overrun	21-30% overrun	31-40% overrun	> 40% overrun
Risk Factors	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
23) Ineffective and inexperienced project manager	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
24) Frequent staff turnover within project team	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
25) Conflicts between users and development team members	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
26) Conflict among users	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
27) Conflicts within development team members	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
28) Excessive schedule pressure and overworked	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
29) Lack of control and coordination within the project	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
30) Overreliance on subcontractor or vendors/suppliers	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
31) Redundancies and overlapping of activities/processes	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
32) Lack of regular reviews against goals	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
33) Large project size	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
34) Tracking of problems within the processes/activities	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
35) Improper sequential of processes/activities	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Appendix H : Questionnaire - Implementation stage

STAGE 5 : IMPLEMENTATION	Likelihood Occurrence					Risk factors impact on cost overrun as % deviation from original estimate				
						Cost				
	None	Unlikely	Likely	Highly Likely	Very highly likely	Very Low	Low	Moderate	High	Very High
						1-10% overrun	11-20% overrun	21-30% overrun	31-40% overrun	> 40% overrun
Risk Factors	1	2	3	4	5	1	2	3	4	5
1) Unsuitable conversion/installation method										
2) Loss of data during conversion/installation										
3) System failure during conversion/installation										
4) Loss of performance during installation										
5) Improper implementation sequence modules/activities										
6) Disruption to existing operation/processes										
7) Difficulty in configuration of system and computer environment/platform										
8) Time constraints in implementation										
9) Large number of interfaces to other system required										
10) Users adaptability to new system										
11) Users lack understanding of system capabilities and limitations										
12) Failure to manage end-user expectations										
13) User resistance to change										
14) Feedback from users not properly analyzed										
15) Time constraints of training										
16) Outlining training schedule										
17) Lack of knowledge and experience of system administrator/configuration manager										
18) Lack of knowledge and experience of implementation team										
19) Ineffective communication between users and implementation team members										
20) Ineffective communication within implementation team members										
21) Changes in management of organisation during implementation										
22) Resources shifted from project due to organisational priorities										
23) Projects affects large number of user departments/units										
24) Inadequate documentation										

Appendix H : Questionnaire - Operation Maintenance stage

STAGE 6 : OPERATION & MAINTENANCE	Likelihood Occurrence					Risk factors impact on cost overrun as % deviation from original estimate				
						Cost				
	None	Unlikely	Likely	Highly Likely	Very highly likely	Very Low 1-10% overrun	Low 11-20% overrun	Moderate 21-30% overrun	High 31-40% overrun	Very High > 40% overrun
Risk Factors	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
1) Lack of organisation's commitment throughout project life	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2) Systems not performing accurately and effectively	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
3) System failure and breakdown	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
4) Inconsistencies of output produced	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
5) Inadequate user documentation	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
6) Poor maintenance schedule	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
7) Poor maintenance procedure	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
8) Lack of technical support	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
9) Threat of hackers	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
10) Viruses/bugs	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
11) Unauthorised user/sabotaj/abuse	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
12) Inadequate safety/security features	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
13) Changes in market condition and organisation priorities	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
14) Systems and programming languages become obsolete	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
15) Actions taken by competitors	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
16) Software not flexible in supporting new requirements and changing user needs	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
17) Cost of training	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
18) Lack of continuous IT investment to sustain competitiveness	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
19) Price fluctuations of hardware and software	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Thank You
End of SECTION C

Appendix I : Risk mitigation strategies

	Effectiveness of risk mitigation strategy in relation to risk reduced						
	Don't know	Not effective	Very slightly effective	Generally effective	Highly effective	Very Highly effective	Exceptionally effective
RISK MITIGATION STRATEGIES	0	1	2	3	4	5	6
1.Define a clear goals and objectives of the project	☐	☐	☐	☐	☐	☐	☐
2.Conduct a through analysis feasibility study	☐	☐	☐	☐	☐	☐	☐
3. Use of project tracking system and regular updating	☐	☐	☐	☐	☐	☐	☐
4. Proper project planning and scheduling	☐	☐	☐	☐	☐	☐	☐
5. Identify critical and non-critical activities	☐	☐	☐	☐	☐	☐	☐
6. Set key performance indicators and standards for stages/processes	☐	☐	☐	☐	☐	☐	☐
7. Lesson learned from past software development projects	☐	☐	☐	☐	☐	☐	☐
8. Identify success criteria	☐	☐	☐	☐	☐	☐	☐
9. Consistent commitment of management	☐	☐	☐	☐	☐	☐	☐
10.Quality control procedure	☐	☐	☐	☐	☐	☐	☐
11. Risk management methodology/techniques	☐	☐	☐	☐	☐	☐	☐
12. Hire external expertise/consultant	☐	☐	☐	☐	☐	☐	☐
13. Contingency plan	☐	☐	☐	☐	☐	☐	☐
14. Conduct pilot testing	☐	☐	☐	☐	☐	☐	☐
15. Prototyping	☐	☐	☐	☐	☐	☐	☐
16. Thorough analysis of development methodology	☐	☐	☐	☐	☐	☐	☐
17. Proper timeframe for testing	☐	☐	☐	☐	☐	☐	☐
18. Conduct a thorough user acceptance test	☐	☐	☐	☐	☐	☐	☐
19. Planned for parallel or phased conversion	☐	☐	☐	☐	☐	☐	☐
20. Developed a clear and detail requirements	☐	☐	☐	☐	☐	☐	☐
21. Incorporate alternative development methodology	☐	☐	☐	☐	☐	☐	☐
22. Backup the system thoroughly	☐	☐	☐	☐	☐	☐	☐
23. Software security checklist and authentication process	☐	☐	☐	☐	☐	☐	☐
24. Cost control procedure	☐	☐	☐	☐	☐	☐	☐
25. Technical support team	☐	☐	☐	☐	☐	☐	☐
26. Proper planning of resources	☐	☐	☐	☐	☐	☐	☐
27. Effective training for staff	☐	☐	☐	☐	☐	☐	☐
28. Effective lines of communication	☐	☐	☐	☐	☐	☐	☐
29. Good project management and leadership	☐	☐	☐	☐	☐	☐	☐
30. Greater degree of users involvement and commitment	☐	☐	☐	☐	☐	☐	☐